

OLE ØYVIND STENSLI

HOVEDFAGSOPPGAVE

HVORDAN KAN STORE ORGANISASJONER ANSKAFFE INFORMASJONSSYSTEMER SOM VIRKER SAMMEN OVER TID?

INFORMASJONSSYSTEMER I STORE ORGANISASJONER, ANSKAFFELSE GJENNOM BRUK AV RAMMEVERK FOR SYSTEMUTVIKLING

Dato 01.05.07

Institutt for Informatikk, Det Matematisk- Naturvitenskapelige fakultet

Universitetet i Oslo

HOVEDFAGSOPPGAVE

INNHALDSFORTEGNELSE

1	INNLEDNING	11
1.1	LESEGUIDE	12
1.2	SAMMENDRAG	12
2	K2IS I HÆREN.....	13
2.1	KORT BESKRIVELSE AV K2IS I HÆREN.....	13
2.2	PROGRAMNIVÅET.....	14
2.3	PROSJEKTENE.....	15
2.3.1	Kommunikasjonsinfrastruktur	15
2.3.2	K2IS Blokk 1	16
2.4	MILEPÆLER I K2IS-UTVIKLINGEN I HÆREN 1997-2005	17
2.5	UTFORDRINGER FOR SYSTEMUTVIKLINGEN – ASPEKTER VED K2IS	18
2.5.1	K2IS som muliggjør organisasjonsutvikling.....	18
2.5.2	K2IS som én av mange aktører.....	19
2.5.3	K2IS som et nettverk uten klare grenser	20
2.5.4	K2IS som mer enn bare teknologi	22
2.5.5	K2IS som del av andre nettverk	23
2.5.6	Helhetlig realisering – hvordan takle utfordringene?	24
3	KRAV TIL METODE FOR SYSTEMUTVIKLING OG INTEROPERABILITET 25	
3.1	KRAV – BEGREPSAVKLARINGER.....	25
3.1.1	Hva er krav?	25
3.1.2	Kategorisering av krav	26
3.2	KRAV TIL EN METODE	27
3.3	HÅNDTERING AV INTEROPERABILITET	27
3.3.1	Interoperabilitet - en introduksjon	27
3.3.2	Agenda for fremtidens forsvar	30
3.3.3	Metode for å håndtere interoperabilitet	31
3.4	HÅNDTERING AV SYSTEMUTVIKLINGEN	31
3.4.1	Krav til systemet	32
3.4.2	Krav til systemutviklingen	32
3.4.3	Anskaffelser og systemutvikling	32
3.5	OPPSUMMERING	33

4	EKSISTERENDE LØSNINGER	34
4.1	SYSTEMUTVIKLING.....	34
4.1.1	Begrepsapparat	34
4.1.2	En tredelt tilnærming	35
4.1.3	Utviklingsmodeller.....	36
4.1.4	Utviklingsmetoder	37
4.1.5	Verktøy	41
4.2	RAMMEVERK.....	42
4.2.1	ISO RM-ODP	42
4.2.2	C4ISR-AF	45
4.3	INTEROPERABILITET	55
4.3.1	Begrepsavklaring	55
4.3.2	Typer interoperabilitet i NATO	57
4.3.3	Modeller for interoperabilitet	59
4.3.4	Initiativ for å oppnå interoperabilitet.....	63
4.4	INFORMASJONSINFRASTRUKTUR.....	64
4.4.1	Hva gjør en infrastruktur irreversibel?	64
4.4.2	Forholdet mellom fleksibilitet og standardisering	65
4.4.3	Hva hindrer så fleksibilitet og evnen til forandring?	65
4.4.4	Konseptet anvendt i utvikling av informasjonsinfrastrukturer!	66
4.4.5	Hvordan II blir irreversibelt.....	66
4.5	IT-STRATEGI.....	66
4.6	ANSKAFFELSESGRUPPE	68
4.6.1	Anskaffelsesregelverk for Forsvaret.....	68
4.6.2	PRINSIX	68
4.6.3	Videre utvikling	68
5	ET RAMMEVERK FOR SYSTEMUTVIKLING.....	69
5.1	ET ARKITEKTURRAMMEVERK	69
5.1.1	Termer og begreper	69
5.1.2	Et rammeverk for arkitekturbeskrivelse	75
5.1.3	En innføring i MACCIS	81
5.2	MACCIS ARKITEKTURMODELLER.....	84
5.2.1	Synsvinkel Virksomhet	85
5.2.2	Synsvinkel Informasjon.....	92
5.2.3	Synsvinkel Behandling	94
5.2.4	Synsvinkel Konstruksjon.....	96
5.2.5	Synsvinkel Teknologi.....	99
6	BRUK AV MODELLER I SYSTEMUTVIKLING	102
6.1	UTVIKLING AV MACCIS	102
6.1.1	Oppsummering etter MACCIS versjon 1	102
6.1.2	Videreutvikling av prosess.....	102
6.2	SYSTEMUTVIKLINGS- OG ANSKAFFELSESPROSESS	103
6.2.1	Virksomhetsmodellering	103

6.2.2	Systemmodellering	109
6.2.3	Modelldrevet anskaffelse	114
7	SYSTEMANSKAFFELSE	121
7.1	INNLEDNING	121
7.2	EKSISTERENDE METODERS ANVENDELSE	122
7.3	MODELLAPPARAT	123
7.3.1	Modellapparat – en helhet	124
7.3.2	Beslutningsmodell	124
7.3.3	Systemmodell	131
7.3.4	Anskaffelsesmodell	135
7.3.5	Modenhetsmodell for anskaffelsesorganisasjonen	142
7.4	OPPSUMMERING	148
7.4.1	Erfaringer etter fase 1 i prosjektet	148
7.4.2	Resultater benyttet videre i K2IS Blokk 1 fase 2	149
8	EVALUERING.....	151
8.1	EVALUERING AV OPPGAVENS BESVARELSE AV KRAVENE	151
8.1.1	Innledning	151
8.2	KRAVET TIL SYSTEMUTVIKLING.....	151
8.3	KRAVET TIL INTEROPERABILITET	152
8.4	KRAVET TIL SYSTEMUTVIKLING OVER TID	152
8.5	HVORDAN KAN STORE ORGANISASJONER ANSKAFFE INFORMASJONSSYSTEMER SOM VIRKER SAMMEN OVER TID?	153
8.5.1	En sammenstilling av besvarelsen i oppgaven	153
8.5.2	Besvarelsen sett opp mot egne krav	153
9	KONKLUSJON - VIDERE ARBEID	155
9.1	KONKLUSJON	155
9.2	VIDERE ARBEIDE – INNFORING I EGEN ORGANISASJON	160
9.2.1	Egen organisasjons modenhet	160
9.2.2	Oppgavens relevans i forhold til egen organisasjon.....	163
10	REFERANSER.....	164

FIGURLISTE

Figur 1 K2IS - en lagdelt fremstilling	13
Figur 2 K2IS - Programnivå og prosjekter	14
Figur 3 Styringsmodellen for Forsvarets virksomhet.	18
Figur 4 Prosjektleders dilemma – avveining mellom tid, kostnad og kvalitet	24
Figur 5 Kravet til Interoperabilitet	29
Figur 6 Kommando og kontrollhjulet	30
Figur 7 De 10 elementene	33
Figur 8 Modell - Metode - Verktøy, et hierarki	35
Figur 9 Modell - Metode – Verktøy	36
Figur 10 Fasene i RUP, visert omfanget for et typisk prosjekt	38
Figur 11 Rational Unified Process – RUP	40
Figur 12 RM-ODP synsvinkler og Software Engineering	43
Figur 13 C4ISR - Arkitekturrammeverk - tre synsvinkler på en arkitektur	46
Figur 14 Overordnet bilde operativ kontekst (OV-1) - Eksempel fra luftvern	47
Figur 15 Systemgrensesnittbeskrivelse SV-1	48
Figur 16 Teknisk arkitekturprofil (TV-1)	48
Figur 17 En sekstrinns prosess for å bygge arkitekturer	50
Figur 18 En felles prosess for å oppnå en interoperabel og kosteffektiv C4ISR dugelighet	53
Figur 19 De manglende elementene for å oppnå en felles prosess	54
Figur 20, Interoperabilitet - Integrert	56
Figur 21 Integrasjon av to systemer til ett	56
Figur 22 Et felles grensesnitt	56
Figur 23 Interoperabilitetsstandarder i NATO	58
Figur 24 LISI Nivå og tilhørende EDB - utrustning	60
Figur 25 Sammenheng mellom NATOs nivå og LISIs nivå	61

Figur 26 ODPs referansemodell	61
Figur 27 NATO C3 Interoperabilitetsmiljø	62
Figur 28 NIMP modell for interoperabilitet	62
Figur 29 - hvorfor standardisere?	64
Figur 30 Modelleringsprosessen: Skapelse av en modell av virkeligheten.	70
Figur 31 Systemnivå	71
Figur 32 Ståsted – Perspektiv - Syn	72
Figur 33 Raffinering	73
Figur 34 Objekter i samspill	75
Figur 35 MACCIS Arkitekturrammeverk	75
Figur 36 Det grunnleggende modellhierarki	76
Figur 37 MACCIS modellhierarki	77
Figur 38 Systemhierarki	78
Figur 39 MACCIS Referansearkitektur for Software komponenter	79
Figur 40 MACCIS systemutviklingsprosess	80
Figur 41 MACCIS utviklingsprosesser og tilhørende modeller	81
Figur 42 Oversikt over MACCIS modellene	82
Figur 43 Avhengigheter mellom arkitekturmodellene	83
Figur 44 MACCIS arkitekturmodeller	84
Figur 45 Synsvinkel virksomhet	85
Figur 46 EV1 Overordnet brukstilfelle	86
Figur 47 Roller og kommunikasjon vist med APP6a symbolikk	86
Figur 48 EV6 Rolle aktivitetsmodell	87
Figur 49 Rolle-aktør	88
Figur 50 Rolle- distribusjon	88
Figur 51 EV3 Business objekter	89
Figur 52 EV3 Business objekter - detaljert	89

Figur 53 Synsvinkel informasjon	92
Figur 54 IV2 Informasjonsstrukturmodell	93
Figur 55 Synsvinkel behandling	94
Figur 56 CV4 Bestanddeler i systemet Felles frekvensadministrasjonssystem (FEFAS/NDFMS)	95
Figur 57 Synsvinkel konstruksjon	96
Figur 58 En-1 Autorisasjon og tilgangskontroll	97
Figur 59 En-2 Distribusjonsmodell	97
Figur 60 En-4 Utplasseringsmodell fra FEFAS/NDFMS	98
Figur 61 Teknologivinkelen	99
Figur 62 TV-1 Tekniske standarder	100
Figur 63 TV-2 Arkitekturutvidelsesmodell	100
Figur 64 TV-4 Komponentteknologiprofil - et eksempel for EJB	101
Figur 65 Konseptuelt, logisk og fysisk skjema	101
Figur 66 Forsvarets overordnede "livssyklus"	104
Figur 67 Overordnet MACCIS prosess	105
Figur 68 Domeneanalysen	105
Figur 69 As-is vs. to-be	106
Figur 70 Kravhåndtering vs. Design	107
Figur 71 Rundgangen mellom krav og design	107
Figur 72 Kravinnhenting	109
Figur 73 Aktivitetene i systemutviklingsprosessen	110
Figur 74 Iterasjoner i software utviklingsprosessen	112
Figur 75 Fasene i software livssyklus	113
Figur 76 MACCIS prosessoversikt	116
Figur 77 Realisering av K2IS Software Subsystemer	117
Figur 78 Livssyklus – Spiralmodellen	118
Figur 79 Unified Process og LCO, LCA og IOC	119

Figur 80 System of Systems med gjenbruk av eksisterende Implementasjoner	120
Figur 82 Leverandør - forvaltningsprosess	122
Figur 83 Modellapparat – systemanskaffelse	123
Figur 84 MBASE Milepæler	125
Figur 85 Forbedringsfirkanten	126
Figur 86 GQM eksempel	127
Figur 87 Sjustegs forretningsplanprosess	128
Figur 88 Nivåer i systemperspektivet	131
Figur 89 Systemhierarki - generisk modell	132
Figur 90 MACCIS - en oversikt over modellarkitekturen	133
Figur 91 - 3 prosesser	135
Figur 92 Anskaffelsesmodell	136
Figur 93 Åpne standardbaserte systemer	137

TABELLER

Tabell 1 Inndeling av krav	26
Tabell 2 Krav til oppgaven	33
Tabell 3 NIPD 6 interoperabilitetsnivå	59
Tabell 4 Virksomhetsvinkelen	85
Tabell 5 Mal for use case	91
Tabell 6 Synsvinkel informasjon	92
Tabell 7 Synsvinkel behandling	94
Tabell 8 Synsvinkel konstruksjon	96
Tabell 9 Teknologivinkelen	99
Tabell 10 Spiralmodellens metodikk, milepæler og faser	119
Tabell 11 Hvordan nå målsettinger i den funksjonelle avklaringen	139
Tabell 12 Hvordan nå målsettinger for arkitektur og leverandørvalg	140
Tabell 13 Hvordan nå målsettinger for risikoreduksjon	141
Tabell 14 Hvordan nå målsettinger for evaluering	142
Tabell 15 Softwareanskaffelse - Dugelighets- og modenhetmodell	146
Tabell 16 Krav besvart i oppgaven	154

1 INNLEDNING

Jeg har tatt for meg dette emnet av flere årsaker. Jeg arbeider til daglig i Forsvarets Logistikkorganisasjon^[FLO]. Jeg begynte i januar 1998 i Hærens Forsyningskommando (HFK) på EDB-materiellkontoret som etter hvert fikk navnet Informasjonssystemkontoret i Forsvarets Logistikkorganisasjon/Materielldivisjonen/Land.

Fra 1. september 2005 ble hoveddelen av kontoret flyttet til FLO/IKT. Jeg legger i hovedsak erfaringer fra perioden før 1. september 2005 til grunn i denne oppgaven. Mine anbefalinger vil forhåpentligvis være relevante for nye FLO. Det å kunne arbeide med en hovedoppgave innenfor samme fagområde som mitt daglige virke har vært en unik mulighet, og ikke minst en stor utfordring.

Min avdelingssjef gjennom mange år Oberst Erik Hammer¹ har gjennom en årrekke arbeidet med modeller for å styre utviklingen av våre kommunikasjons- og informasjonssystemer. I denne perioden hadde han et samarbeid med SINTEF for å utvikle et sett av arkitekturmodeller. Dr Ing Arne Jørgen Berre har vært faglig leder for denne gruppen.

Jeg har også vært student ved Institutt for Informatikk. Her har jeg i særlig grad blitt inspirert gjennom fagene informasjonsinfrastruktur og modellering med objekter, der henholdsvis Ole Hanseth og Arne Jørgen Berre har undervist.

HFK har i denne perioden gjennomgått en omstilling, der en ny organisasjon Forsvarets Logistikkorganisasjon, FLO er etablert. Den nye organisasjonen FLO fikk sin siste utforming 1. september 2005.

Fra 1. september 2005 består FLO av 3 hovedenheter som her er relevante, dette er Systemstyring, Investering og IKT. FLO/Systemstyring har tatt inn alle relevante elementer knyttet til forvaltning av materiell kalt systemansvar. FLO/Investering er skilt ut som egen enhet for å forestå alle større anskaffelser i Forsvaret. FLO/IKT har samlet all IKT-faglig virksomhet.

Nytt materiell skal anskaffes etter bestemte rutiner. Materiellet skal fremskaffes slik at kostnadene blir holdt nede på et minimum ved anskaffelsen. Dette er langt på vei en merkantil sak, som best ivaretas ved bruk av anbud og åpen konkurranse. FLO/S og IKT er ansvarlig for drift og vedlikehold i hele materiellets levetid. Anskaffelse er altså ikke bare initialanskaffelse, men inkluderer også drift og vedlikeholdsmessige aspekter. Anskaffelse i FLO er altså mer enn godt merkantilt håndverk. Hvordan skal man forutse slike utgifter og er det særlige egenskaper ved informasjonssystemer som må ivaretas?

HFK inngikk en kontrakt med leverandøren CCIS House om leveranse av et kommando og kontroll informasjonssystem – første gang innen medio år 2002. CCIS House har rolle som leverandør i kontrakten, men fordi selskapet er et samarbeidsforum for flere selskaper, har det største selskapet, Kongsberg Defence & Aerospace tatt rollen som hovedleverandør.. Leveranser fra avtalen har fortsatt videre, og siste avtalte leveranse er nå i juni 2007. FLO revurderer i disse dager rammeavtalen med CCIS House.

Forsvaret er midt i en større omstilling. Denne omstillingen kalt transformasjon i Forsvaret, innbefatter også Nettverksbasert Forsvar. ”En sentral del av transformasjon er implementering av Nettverksbasert Forsvar (NBF), som i korthet går ut på å utnytte informasjonsteknologi til å organisere Forsvarets operative struktur og enheter i et samvirkende nettverk.”^[ST-ppp42]

¹ Ob Erik Hammer gikk av med pensjon første kvartal 2005.

Omstillingen i Forsvaret har så langt omsatt denne tanken om NBF ved to organisatoriske trekk. Det ene er ved etableringen av Forsvarets kompetansesenter for kommando og kontroll informasjonssystemer – FK KKIS^[FK-KKIS]. Det andre er ved etableringen av nye FLO, der presumptivt alle kommando- og informasjonssystemer samles i nye FLO/IKT.

FLO står dermed overfor store utfordringer knyttet til kommando- og kontrollsystemer. Uavhengig av min egen plass i denne organisasjonen, tror jeg problemstillingen i oppgaven er mer enn relevant også i dag.

1.1 LESEGUIDE

- Et faktisk prosjekt - K2IS i den norske Hæren [kapittel 2].
- Krav til Systemutviklingen og kravet til interoperabilitet [kapittel 3].
- Eksisterende løsninger og metoder [kapittel 0] med fokus på militære standarder.
- Et rammeverk for systemutvikling [kapittel 5] basert på MACCIS.
- Anbefalt systemutviklingsprosess [kapittel 6]
- Forslag til helhetlig systemanskaffelse [kapittel 7]
- Evaluering - av besvarelsen opp mot de krav er beskrevet i kapittel 2 [kapittel 8]
- Konklusjon – og mulig videre arbeide (Evaluering av egen organisasjon) [kapittel 9]

1.2 SAMMENDRAG

Jeg har i denne oppgaven tatt for meg problemstillingen; hvordan kan store organisasjoner utvikle systemer som virker sammen over tid?

I denne oppgaven har jeg sett på hvordan Forsvaret gjennomførte en større anskaffelse av IKT samtidig som man innførte en ny metode for å styre og lede slike anskaffelser.

Forsvaret bestilte et rammeverk, MACCIS fra SINTEF Tele & Data. Rammeverket blir presentert i denne oppgaven. Dette rammeverket ble i svært liten grad tatt i bruk av prosjektet, og det ble ikke stilt eksplisitte krav til leverandøren om at systemet skulle beskrives i henhold til MACCIS.

Det var flere grunner til at dette ble gjort, resultatet er at vi bare fått begrensede erfaringer i å benytte MACCIS til anskaffelse av informasjonssystemer.

Jeg hevder at vi i tillegg til å ha fokus på rammeverket MACCIS også burde sett på beslutningsprosesser og modenhet i egen organisasjon. Jeg tilfører derfor modellapparatet til MACCIS en beslutningsmodell og en modenhetsmodell.

Vi kunne ha innført MACCIS gradvis, og dermed gjennomført en inkrementell og iterativ innføring også av modeller og arkitektur som metode. På denne måten mener jeg at en stor organisasjon som Forsvarets Logistikkorganisasjon i større grad vil kunne gjennomføre anskaffelser av store komplekse systemer, som vi forventer skal virke sammen over tid.

Jeg har i denne oppgaven sett på noen av de fallgruver man kan gå i ved anskaffelse av informasjonssystemer i store organisasjoner. Jeg håper at noen av mine observasjoner og forslag til forbedringer kan være nyttig når vi nå skal gjennomføre tilsvarende prosjekter også i fremtiden.

2 K2IS I HÆREN

2.1 KORT BESKRIVELSE AV K2IS I HÆREN

Dette kapitlet skal gi en kort innføring og beskrivelse av Hærens Kommando og Kontroll Informasjonssystem(K2IS).

K2IS er ikke noe endelig system - heller en visjon, eller et sett av systemer - K2IS er tenkt som en helhet av systemer – et systemprogram. K2IS i Hæren bygde på en samling erfaringer. På den ene siden fra et tidligere prøveprosjekt - Operativ EDB - der hovedkonklusjonen var at informasjonsteknologi vil være et meget stort aktivum innenfor Kommando og Kontroll, og det anbefales å fortsatt investere i K2IS. På den annen side en serie studier fra Forsvaret Forskningsinstitutt (FFI) på nytt ildledningssystem (ODIN 2) til Hæren og KKI-Hær.

Alle informasjonssystemer er avhengig av kommunikasjon – så også K2IS. I Hæren har man gjennom en lengre periode fremskaffet et taktisk digitalt kommunikasjonssystem (TADKOM). Dette systemet er blitt utvidet med et meldingssystem (TMHS) og har etter hvert også blitt utvidet med en ny digital radio (MRR). Denne infrastrukturen med TADKOM som basis er i stor grad en premissgiver for K2IS. Ved starten av prosjekt K2IS - år 2000 - er kommunikasjonsinfrastrukturen for en stor del etablert, og kommunikasjonsprosjektene er ferd med å nærme seg terminering.

K2IS skal brukes av enkeltmann om han er til fots, sitter i kjøretøy, kommandoplasscontainer eller i klasserom for opplæring. En viktig del av K2IS er derfor integrering og tilpassning til det fysiske miljø.

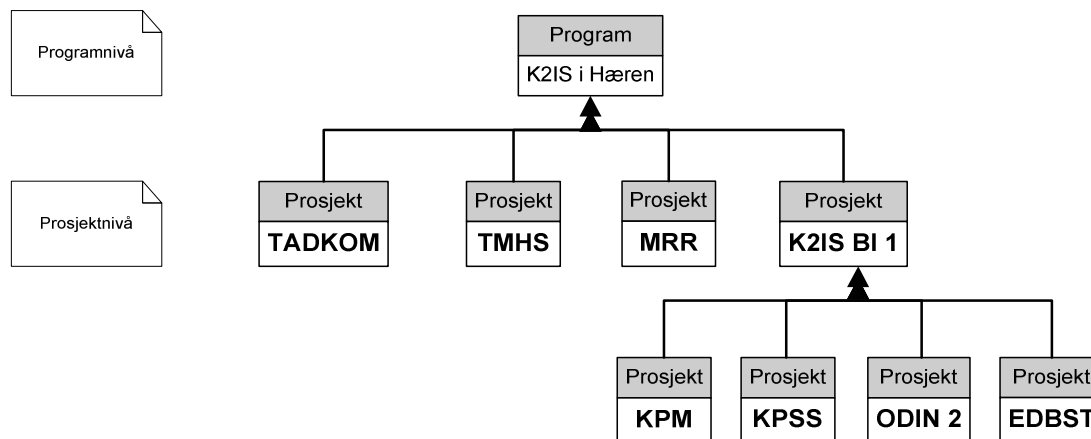


Figur 1 K2IS - en lagdelt fremstilling

Ettersom K2IS brukte svært lang tid på realisering oppsto det et vakuum hos brukerne. Det ble stilt krav fra brukerne om å få et system i mellomtiden. Systemet HærTas, et funksjonelt system utviklet av Teleplan AS, ble tatt i bruk for et øvingsår 2000/2001. Øvingsåret etter 2001/2002 ble K2IS tatt i bruk.

2.2 PROGRAMNIVÅET

Programmdirektivet ^[PROG97] legger opp rammene for utvikling av K2IS. Direktivet ble første gang utgitt i 1997, og ble senere oppdatert i 1999. Programmdirektivet legger opp en langsiktig men ambisiøs plan for gjennomføring av prosjekter. Programnivået fungerer som en organisatorisk overbygging over prosjektene, for å sikre en helhet. Dette prinsippet synliggjorde tidlig at selv om prosjektene i forsvaret er selvstendige i betydning krav til leveranser og finansiering, vil det likevel være behov for styring ut over prosjektets rammer. Gruppering i program – der prosjekter som har visse likheter samles – er et prinsipp som i dag er utvidet til også å omfatte alle andre prosjekter i forsvaret. Fra 2005 er alle prosjekter i forsvaret organisert innenfor 18 utvalgte programområder. Fra 2007 er dette redusert til 7 programområder, Organiseringen av K2IS i Hæren som et program var i så måte et eksempel til etterfølgelse.



Figur 2 K2IS - Programnivå og prosjekter

Hæren var fra 1997 gjennom en kraftig nedskjæring på investeringssiden. Første versjon av programmdirektivet i 1997 ble signert av Generalinspektøren for Hæren og publisert gjennom Forsvarets kanaler. En oppdatert versjon er laget, men er ikke distribuert tilsvarende, og det har siden ikke blitt laget noe tilsvarende direktiv. Gyldigheten av direktivet har derfor gradvis blitt redusert.

2.3 PROSJEKTENE

Her følger en kort gjennomgang av de hovedprosjektene som har vært gjennomført innenfor Hærens K2IS.

2.3.1 KOMMUNIKASJONSINFRASTRUKTUR

Disse prosjektene er for en stor del realisert som en enhetlig systemløsning innenfor laget kommunikasjonsinfrastruktur.

2.3.1.1 TADKOM

Taktisk Digital KOMmunikasjonssystem - TADKOM er Hærens taktiske, mobile kommunikasjonssystem^[Systbesk] ^[TADKOM].

Systemet tilbyr de tjenester som er vanlig i et digitalt telefonsystem, tilsvarende sivile systemer som for eksempel ISDN for tale/data. Systemet består fysisk av et antall kjøretøy som kobler seg sammen til et helhetlig system ved hjelp av radiolinje. Til hvert kjøretøy kan et gitt antall abonnenter tilknytte seg.

Kjernen i systemet er en digital og meget kompakt Datastyrt Feltveksler som skal fungere i rollen både som trunksvitsj og aksessvitsj med integrert overføring av tale og data. Hver feltveksler tilbyr pakkesvitsj, radio aksess, gatewayfunksjoner og integrert bulk krypto. Hver enhet har en autonom funksjon i nettet, og tilbyr basistjenester etter EUROCOM standarden. For tilknytning av både analoge og digitale telefoner benyttes en multiplekser. Den innebygde pakkesvitsjen i feltveksleren gir brukeren aksess til et fullt integrert datanettverk basert på X-25.

Som transmisjonsbærer for områdenettet benyttes primært radiolinje. Det benyttes to typer radiolinje i områdenettet. Disse opererer i henholdsvis SHF og UHF båndet.

I tillegg disponerer Forsvaret et landsdekkende stasjonært strategisk kommunikasjonssystem - FDN. For overgang til slike andre systemer har TADKOM et utvalg flerkansals gatewayfunksjoner tilgjengelig. FDN/TADKOM GW er primært for nasjonalt bruk, men er basert på ISDN PRA som er en åpen internasjonal standard for PCM systemer. I tillegg er US/EUROCOM GW (STANAG 4206) og EUROCOM GW tilgjengelig for bruk mot allierte nasjoners taktiske nett.

2.3.1.2 TMHS

Taktisk Meldingshåndteringssystem (TMHS) er forlengelsen av meldingssystemet i det strategiske nettet FDN ut i feltforhold.

Det strategiske meldingssystemet Meldingstjenesten i Forsvaret er et stasjonært landsdekkende system for utveksling av elektroniske meldinger, basert på internasjonal standarder X.400, NATO STANAG 4406 MMHS og NATO ACP 127. I tillegg til samme grunnfunksjoner som kommersielle meldingssystemer er systemet opprinnelig designet som et Multi-Level Secure system tilpasset militære behov.

TMHS automatiserer meldingstjenesten, men gir også et standardisert grensesnitt for utveksling av meldinger. Dette innebærer at TMHS må fungere på alle aktuelle transmisjonsmedier, samt kunne håndtere en blanding av gradert og ugradert meldingstrafikk. Systemet må være kompatibelt og integrerbart med K2IS.

Systemet TMHS har foruten de elementer som inngår i kommunikasjonsinfrastruktur et element av logisk distribusjon og en meldingsklient med stridsjournal – en applikasjon.

2.3.1.3 MRR/LFR

Kommunikasjonssystemet TADKOM er utvidet med Multi Rolle Radio som er Forsvarets nye generasjon taktiske radiosystem^[MRR]. Lett FlerbruksRadio, LFR er en ytterligere miniatyrisering av MRR. MRR gir brukeren stor fleksibilitet ved at samme radiosett kan benyttes i flere roller. MRR er godt egnet til overføring av data med strenge krav til ytelse samtidig med god beskyttelse mot Elektronisk Krigføring. MRR er et tjenesteintegreert digitalt radiosystem, med rolle som nettradio, pakkeradio og mobiltilknytning. I rollen som nettradio kan brukeren benytte MRR i egne nett som også kan virke mot tradisjonelle VHF-radiosett. I rollen som pakkeradio kan MRR brukes for overføring av data i henhold til X-25. Dette benyttes i dag innenfor luftvernssystemet samt for overføring av tekstbaserte meldinger til og fra sluttbrukere. I rollen som mobiltilknytning har brukeren av MRR mulighet for å kommunisere mot TADKOM på tilsvarende måte som innenfor GSM mobiltelefoni.

For å skape gjennomgående god sikkerhet for hele områdesambandet er det integrert krypto i både trunknettet og MRR.

2.3.2 K2IS BLOKK 1

2.3.2.1 Mange parallelle initiativ

Hæren arbeidet gjennom en årrekke med å ta fram informasjonssystemer for bruk under feltforhold. Frem til 1997 hadde dette ikke ført til en realisert løsning – en årsak til dette mente man var mange parallelle initiativ uten en kraftsamling.

2.3.2.1.1 ODIN 2

Ildledningssystemet ODIN 2 hadde som mål å tilføre Feltartilleriet et helhetlig system for ildledning, kommando og kontroll. Dette var et helhetlig system som berørte alle elementene i den lagdelte Figur 1 K2IS - en lagdelt fremstilling.

2.3.2.1.2 EDBST

EDB Stabsstøtte – EDBST skulle trinnvis anskaffe et system som behandler, utvikler og presenterer informasjon for med dette å støtte stabsarbeidet i feltavdelingene. Prosjektet skulle ta fram en basis plattform som sikrer interoperabilitet mellom våpensystemene og ledelsessystemene på divisjonsnivå og lavere, samt med høyere nivå. Prosjektet hadde som målsetting å ta hensyn til nødvendig organisasjonsutvikling og eventuelle endringer av prosedyrer slik at verktøyet kunne bidra til å forenkle og effektivisere stabsarbeidet. EDBST skulle presentere situasjonsbildet for å bidra til situasjonskontroll. EDBST hadde foruten elementer av informasjonsinfrastruktur primært fokus på applikasjonslaget.

2.3.2.1.3 KPM

Kommandoplass materiell – KPM var en nødvendig forutsetning for å integrere EDBST i alle ønskede kommandoplasser. KPM var kun rettet mot det nederste laget i Figur 1 K2IS - en lagdelt fremstilling.

2.3.2.1.4 KPSS

Kommandoplass sambandssystem – KPSS skulle kunne betjene EDB-støttesystemer internt i kommandoplasser, samt sørge for intern talekommunikasjon i disse kommandoplassene. KPSS skulle være forlengelsen og integrere kommandoplassene med TADKOM på taktisk nivå. KPSS skal knyttes opp mot FDN og sivile nett på høyere nivå, primært igjennom TADKOM.

2.3.2.2 K2IS Blokk 1

De mange prosjektene som til dels hadde overlappende målsettinger, ble ikke vurdert å være farbar realisering av K2IS i Hæren. I 1997 gjennomførte utvalgte deler av norsk forsvarindustri en mulighetsstudie IG-studien for å vurdere om det var mulig å gjennomføre prosjektene ODIN 2, EDBST, KPSS og KPM som en helhetlig satsing.

Våren 1998 inngikk Hærens Forsyningskommando en kontrakt med CCIS-House som skulle være den første av flere kontrakter, felles skulle være målsettingen om å ta fram et kommando og kontrollsystem for bruk i taktiske enheter. K2IS har som hovedmål å understøtte avdelingssjefer og feltstaber i utøvelsen av sine funksjoner. For å oppnå maksimal utnyttelse av K2IS er det viktig at de kan utveksle informasjon på en enkel og hurtig måte. Det vil være en utfordring i K2IS å benytte TADKOM som kommunikasjonskanal på en mest mulig effektiv måte. K2IS vil være svært avhengig av TADKOM.

2.4 MILEPÆLER I K2IS-UTVIKLINGEN I HÆREN 1997-2005

- ”Strategisk samarbeid for Hærens K2IS” – kontrakt som regulerer forholdet mellom Forsvaret og CCIS House AS, når det gjelder å ta fram K2IS elementer i Hæren, godkjent av Det Kongelige forsvarsdepartement (FD)1997-04-14.
- IG – studien, 1997
En studie med fokus på ODIN II og enkelte kommunikasjonsløsninger.
- K2IS Blokk 1 - fase 1. 1998-2000
En første gjennomgang av K2IS fra konsept til definisjon. Avsluttet med en teknisk systembeskrivelse av K2IS
- Hovedkontrakt for K2IS Blokk 1
Ble godkjent 26. april 2000 med CCIS House AS som leverandør. Hovedansvarlig i CCIS House var Kongsberg Defence and Aerospace (KDA).
- Utpøving av HærTas i 6. Divisjon
Ettersom K2IS har brukt svært lang tid på å gå fra konsept- og definisjonsfase oppstår det et vakuum hos brukerne. Dette har medført at det har blitt innført et system i til utpøving. HærTas er et funksjonelt system utviklet av Teleplan AS, en aktør som står utenfor CCIS House, men som har lang erfaring med leveranser til Forsvaret. De er blant annet leverandør av K2IS for det strategiske nivå i Forsvaret – NORCCIS II.
- Leveranse til øvingsåret 2001/2002 av K2IS Blokk 1
- Årvisse leveranser fra 2003 frem til siste planlagte leveranse i juni 2007.

2.5 UTFORDRINGER FOR SYSTEMUTVIKLINGEN – ASPEKTER VED K2IS

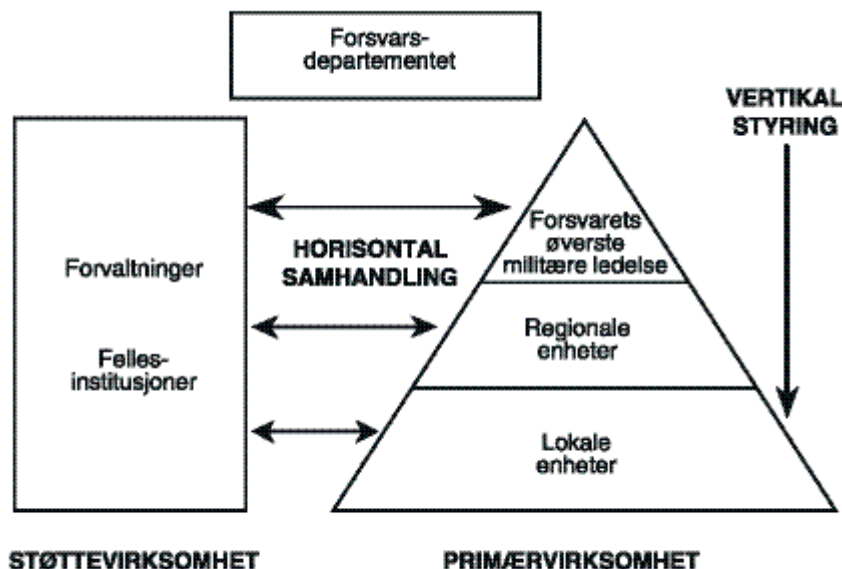
Jeg har siden jeg første gang skrev dette kapitlet hatt en utfordring med å beskrive hva K2IS er. Jeg har ønsket å bruke K2IS som et eksempel – et "case" – fordi jeg oppfatter at nettopp K2IS har i seg svært mange utfordringer for en systemutvikler.

Jeg tar her for meg noen aspekter som jeg mener systemutviklingen av K2IS må være særlig oppmerksom på. Inndelingen er ikke helt tilfeldig, den er inspirert av faget IN-IS, mer spesielt hvordan forstå informasjonsinfrastruktur^[IIS-bok].

2.5.1 K2IS SOM MULIGGJØR ORGANISASJONSUTVIKLING

Prosjekt EDBST sin målsetting skulle ta hensyn til nødvendig organisasjonsutvikling og eventuelle endringer av prosedyrer slik at verktøyet kunne bidra til å forenkle og effektivisere stabsarbeidet. Det var ikke tilfeldig at man forutså endring i organisasjon og prosedyrer. Forsvaret var i ferd med å gjennomgå fundamentale endringer nærmere 10 år etter Berlinmurens fall. St.meld. nr. 22 (1997-98)^[St-meld22] omtaler Forsvarets struktur og virksomhet slik:

- Overgangen til et manøverorientert operasjonskonsept har også en viktig betydning for Forsvarets struktur og virksomhet. En slik måte å operere på krever evne til å føre samordnet strid i høyt tempo og med høy intensitet.
- Forsvarets styringskonsept ...består av en styringsmodell og fire ledelsesprinsipper. Styringsmodellen skiller mellom primær- og støttevirksomhet. Primærvirksomheten skal sørge for at Forsvaret løser sine oppgaver, mens støttevirksomheten omfatter verksteder, forvaltning, fellesinstitusjoner mm, som leverer varer og tjenester til primærvirksomheten. Styringen er delt i en vertikal og en horisontal dimensjon.



Figur 3 Styringsmodellen for Forsvarets virksomhet.

Styringskonseptets ledelsesprinsipper er innrettet mot lederatferd på alle nivåer og i alle deler av organisasjonen, og består av:

- klargjøring av ansvar og myndighet
- delegering av oppgaver og myndighet så langt dette er hensiktsmessig
- ansvar og myndighet følges ad
- ansvar gjøres gjeldende

„

Dette er ambisiøse mål både for hva Forsvaret skal levere, og ikke minst for hvordan Forsvaret skal drive sitt daglige virke målrettet og effektivt. Det er ikke til å undre seg over at mange, også jeg selv ble meget inspirert av dette.

En mer manøverorientert tilnærming forutsetter en raskere beslutningssløyfe – en mer effektiv utøvelse av Kommando og Kontroll. Det er altså en forutsetning at K2IS skal være en sterk bidragsyter for å effektivisere Kommando og Kontroll. En effektivisering kan naturlig nok ta forskjellige former, det kan være prosedyremessige gevinster som gir økt tempo, men man er ikke fremmed for at det kan oppnås organisatoriske gevinster – i form rasjonaliseringer eller strukturelle endringer.

Beskrivelser av anvendelsesområdet til K2IS – domenet K2 blir da en interessant faktor. Her skal man ta utgangspunkt i å understøtte virksomheten slik den er i dag, samtidig skal man være åpen for at det kan bli endringer. Endringene skal ikke forskutteres, men man har en klar formening om et effektiviseringspotensiale. Informasjonssystemet som skal støtte opp under denne virksomheten må da lages på en slik måte at det kan støtte den eksisterende organisasjon så vel som en fremtidig organisasjon.

K2IS skal altså være en underliggende og muliggjørende funksjon for en mer fundamental utvikling av organisasjonen.

2.5.2 K2IS SOM ÉN AV MANGE AKTØRER

Utvikling i Hæren har tradisjonelt vært todelt mellom styrkeprodusenten og forvaltningen. Styrkeprodusentens ansvar er på den ene siden å produsere fullverdige avdelinger og på den andre siden å være kravstiller til forvaltningen for de tekniske systemene.

2.5.2.1.1 Tradisjonelt

I Hæren har virksomheten tradisjonelt sett vært faglig delt mellom inspektører for forskjellige funksjonsområder. Dette vil si at det er en inspektør for artilleri, en for samband, en for logistikk osv. Deres jobb består i hovedsak å produsere funksjonelle enheter på vegne av Generalinspektøren for Hæren, slik at Hæren blir forberedt for de oppgaver den skal ha. Dette kalles styrkeproduksjon.

Avdelingene som trener soldater og befal for Felthæren har vært stedet der inspektøren leverer sin kapasitet. Avdelingene skal på tilmålt tid bringe enkeltmennesker fra utrenet personell til godt samspillende funksjonelle lag, tropper og større enheter. De vil i mange situasjoner ha nok å fordreive tiden med. Med et godt samspill mellom inspektør og de nevnte avdelingene har derfor inspektørene stått for planlegging av nye kapasiteter, nytt materiell og nye prosjekter.

For at en enhet skal ha den kapasitet som er ønsket, innbefatter dette ofte utstyr som forvaltningen bidrar med å fremskaffe. Inspektørene er da kravstillere, eller kunder, og forvaltning i dette tilfellet Hærens Forsyningskommando leverandør.

Hærens Forsyningskommando har i forbindelse med anskaffelse av materiell i stor grad hatt en inspektør å forholde seg til.

Med en stabil organisasjon med klart definerte oppgaver har denne måten å drive fremskaffelse av materiell vært formålstjenlig.

2.5.2.1.2 Det nye Forsvaret

I løpet av de siste 15 årene har Hæren gått fra å bestå av 13 Brigader pluss territorielle regiment, forberedt for oppdrag i Norge, til å bestå av mellom 1 og 3 Brigader med fokus på internasjonale operasjoner. I denne perioden gjøres også Hæren mer mobil det vil i praksis si at hovedkomponenten, Infanteriet, i større og større grad skal benytte diverse kjøretøy som en del av sine lag og tropper. Det nye Forsvaret er i støpeskjeen og er ikke lenger en stabil organisasjon med kjente oppgaver.

Et fokus på internasjonale operasjoner endrer også prioriteringen og hvem som skal lede an i prosjektene. Tempoet i omstillingen har gjort at i enkelte tilfeller prøver man ut nye arbeidsformer i avdelingene, uten at inspektørene leder an i arbeidet.

2.5.2.1.3 Utfordring

Anskaffelse av materiell innenfor området Kommando og Kontroll innbefatter alle inspektører. Når man i tillegg har en organisasjon i støpeskjeen samtidig som antallet planlagte prosjekter er svært mye større enn bevilgende midler kan det til tider være en utfordring å finne en representativ kunde til K2IS.

For å lykkes med et K2IS prosjekt vil det være viktig å sette prioriteringen riktig i organisasjonen. Brukermedvirkning betyr ikke at alle enkeltpersoner skal få påvirke like mye. Det må gjøres et utvalg – det er de riktige representantene for sluttbrukerne som skal påvirke. Det er da en utfordring å få til en god avklaring mellom de riktige inspektører og et utvalg av sluttbrukere fra avdelingenes.

K2IS er altså delt mellom mange brukere og flere brukergrupper

2.5.3 K2IS SOM ET NETTVERK UTEN KLARE GRENSER

K2IS er i utgangspunktet avgrenset til Hæren. Selv om Hæren slik vi har sett er i stadig utvikling, er det tross alt en avgrenset enhet. Likevel er Hærens enheter – eksempelvis brigaden historisk sett en operativ enhet sammensatt av flere våpenarter, som bare delvis kan operere selvstendig^[LANDOPS].

2.5.3.1 Nettverk av interessenter

K2IS som initiell kapasitet ODIN 2 og EDBST skal understøtte deler av brigadens virksomhet og avdelinger. Hvordan er det da med de andre enhetene? Hvordan vil utrulling til enkelte avdelinger påvirke spillet mellom enheter og dermed helheten? Det er ikke til å legge skjul på med K2IS som en driver for å videreutvikle og omstille den operative evnen, er interessen stor for å skaffe K2IS hos den enkelte avdeling. Utviklingen av Hæren er et spill mellom mange interesser og interessenter. En stadig større fokus på husholdning innenfor begrensede økonomiske rammer har ikke gjort den interne prioriteringen av investeringsmidler mindre.

I prinsippet kan vi fastslå at foruten økonomiske begrensninger som ligger i realiseringen av K2IS, er det tilnærmet ingen grense for antall brukere og noder som inngår i K2IS.

2.5.3.2 Nettverk av leverandører

Historisk har forsvarsteknologi fram til midten av nittenåttitallet ligget teknologisk langt fremme. I de siste tiår har den generelle teknologiske utvikling gått fortere og fortere. I samme periode har bevilgningene til forsvarsformål blitt redusert. Resultatet har blitt at kommersiell teknologi langt på vei har tatt igjen forsvarsteknologi, og på enkelte områder gått forbi. Alle nasjonene i NATO har som en naturlig konsekvens begynt å se hvor og når det er mulig benytte COTS^[COTS] materiell.

Tradisjonelt har Forsvaret hatt et utvalg leverandører som har spesialisert seg på leveranser til Forsvaret. Innenfor K2IS er ikke dette noe unntak – og den strategiske samarbeidsavtalen viser at dette ønskes videreført. De tidligere leverte systemer, da særlig kommunikasjonssystemer utgjør en slik formidabel investering at Forsvaret må sikre seg fortsatt tilgang til kompetanse for å understøtte driften i materiellets levetid. Den kompetanse som her kan gjøres tilgjengelig vil Forsvaret anvende som basis for å videre utvikle K2IS.

Når vi nå ønsker å benytte COTS i større grad, vil dette kunne få flere effekter.

Den mest umiddelbare effekten er selvfølgelig innsparinger på økonomiske midler. Slikt omtales i Forsvaret som at ”det blir mer Forsvar for hver krone”. Med samme budsjetttramme vil materiellet kunne oppgradere oftere ved å anvende COTS. På den måten kan også det materiellet som Forsvaret benytter bedre følge med i den tekniske utviklingen. Bruk av COTS kan også bety at man må gi avkall på noe. Forsvaret er tradisjonelt kjent for å stille tøffe krav til materiellet. Dette er for eksempel krav til å tåle utebruk. Det blir vanligvis stilt krav til at materiellet skal tåle bruk under ekstreme temperaturer, så vel som det stilles krav til at materiellet skal tåle fuktighet og vibrasjoner. Dette er andre krav enn det som volummarkedet – forbrukerelektronikken ellers i samfunnet retter seg inn mot.

Det er primært her de tradisjonelle leverandørene til Forsvaret har hatt sin nisje.

Hvordan ivaretas balansen mellom utnyttelse av kompetanse i de utvalgte bedriftene og den generelle utviklingen innenfor det kommersielle markedet?

De ikke-funksjonelle krav – f. eks robusthet, legger begrensninger på hvor mye COTS som kan benyttes. Det må kontinuerlig gjøres avveininger om hva som er godt nok. Tradisjonelle måter å tenke på og forestillinger er i endring, og det kan hevdes at det er ingen grense for antall leverandører som har interesse for K2IS.

2.5.3.3 Nettverk – fysisk infrastruktur

TADKOM er valgt som bærer av K2IS. TADKOM er basert på protokoller fra OSI-verdenen. Dette er protokoller som er svært kraftige hva gjelder funksjonalitet. De tekniske løsningene er svært robuste. Denne robusthet er blitt en del av de krav som man er vant til å stille. TADKOM har gjennom sin utvikling etter hvert blitt satt i stand til å gjøre alt man kan forvente av et sambandssystem. Skulle man dog tenke seg for eksempel å anvende internett-teknologi på toppen av et slikt system, vil man måtte gjøre et brudd med de tekniske muligheter som ligger der. Når jeg vinkler det som eksisterer i dagens TADKOM positivt, er det fordi jeg mener at denne teknologien er muliggjørende. Det er samtidig klart at teknologien er krevende. Dette ser vi ikke minst når TADKOM skal taes i bruk. Det krever betydelige ressurser og spesialkompetanse. Mer moderne teknologi er enklere å anvende. TCP/IP for eksempel er utrolig mye enklere som protokoll enn X-25. Samtidig finnes det flust av brukervennlige applikasjoner for TCP/IP, jeg kjenner ikke mange tilsvarende for X-25. Det at man har valgt X-25 kan gjøre det vanskelig å tenke seg å kjøre TCP/IP på toppen. Dette vil umiddelbart oppfattes som å kjøre

^[COTS] Commercial Of The Shelf.

to "tilsvarende" transportløsninger oppå hverandre. Derfor kan man lett bli tro mot de tidligere løsninger og fortsette egenutvikling basert på dette. Dette kan fort bli en lock-in situasjon. Man er bundet av tidligere løsninger og ser ingen mulige forbedringer. Dette fenomenet at man ikke greier å frigjøre seg fra tidligere valg, kan også beskrives som "path-dependence"². Man har valgt en vei, i dette tilfellet OSI-veien, og da må man fortsette den samme veien videre. Dette til tross for at det vil utvikles smartere og billigere løsninger for TCP/IP, men slike løsninger blir ikke tilgjengelig på grunn av tidligere teknologiske valg. Her er det en spennende avveining mellom bruk av de tekniske løsninger man så for seg for 10 år siden og som det er benyttet betydelige midler for å fremskaffe og de tekniske løsninger som i dag kan skaffes som del av en hyllevareanskaffelse til en dertil hørende svært lav pris. Innefor K2IS har vi i dag og vi kommer til å ha dette som et spenningsfelt mellom nytt og noe gammelt – det er ingen grense for antall noder og nettverk som inngår i K2IS.

2.5.3.4 Nettverk - logisk infrastruktur

K2IS skal anvendes i et taktisk miljø. Karakteristiske egenskaper er enheter som er i bevegelse, det er stor mobilitet. Dette oppfattes som å være særegent. Hvorvidt det er det kan sikkert diskuteres - hvis man ser på de tjenester som etter hvert gjøres tilgjengelig på det kommersielle marked kan man sikkert finne andre brukere som har tilsvarende behov. F. eks. en tjeneste som "mobilt kontor", som skal gjøre det mulig å utføre tjenester som man tradisjonelt måtte sitte i ro ved kontorpulten for å utføre, men som man nå skal kunne utføre men man er i bevegelse. Dette er et eksempel på dynamikk. Et mer spesielt krav til forsvarsmateriell er at man ikke kan se bort fra at dette er tenkt benyttet under forhold som kan medføre tap. Enheter kan bli slått ut. Dette kan være sentrale elementer som rett og slett ikke vil kunne utføre sine tjenester lengre. Da må andre enheter kunne ta over. Dette er dynamikk i en annen forstand – der enhetene selv må kunne overta hverandres roller. Kravene til dynamikk stiller krav til sambandssystemene når det gjelder mobilitet. Den andre type dynamikk stiller krav til hvordan man lagrer og distribuerer informasjon. Informasjon som er kritisk for virksomheten bør vurderes lagret mer enn et sted. Dette medfører at informasjonsinfrastrukturen må bygges svært robust. Alle alternativer for lagring, ruting og fremhenting av informasjon må kunne håndteres.

Det kan derfor sies at det er ingen grense for antall logiske noder og logiske nettverk som inngår i K2IS.

2.5.4 K2IS SOM MER ENN BARE TEKNOLOGI

Et av de trekkene som er særegne for Norge i forhold til våre alliansepartnere i NATO, er at vi så langt har basert vår forsvarsevne på et mobiliseringsforsvar. Dette innebærer at vårt forsvarsmateriell i stor grad vil benyttes av andre enn de stadig tjenestegjørende offiserer.

Vårt mobiliseringsforsvar er basert på at baker Hansen, snekker Nilsen og direktør Pettersen skal møte opp ett eller annet sted og være troppssjef Hansen, sambandstekniker Nilsen og geværmann Pettersen henholdsvis. Hva skal til for at disse tre skal kunne anvende et system som er spesiallaget? Kan man lære opp noen som ikke har vært inne på repetisjon på 5 år hvordan man skal sette opp brukeraksstabeller i K2IS? Det er en utfordring å fremskaffe systemer som er lette å bruke, eventuelt opplæringsprogram som raskt setter nye brukere i stand til å utføre oppgaver ved hjelp av ny teknologi.

² Path-dependence; i.e. passed events will have large impacts on future development and in principle irrelevant events may turn out to have tremendous effects. Se forøvrig [IIS-bok]

Et vesentlig poeng er her å utnytte den kunnskapen den enkelte har ervervet seg i sin daglige jobb. Ved bruk av andre informasjonssystemer har den enkelte fått en oppfatning av hvordan slike systemer skal brukes. En stor grad av "look-alike" med gjenkjennbar oppførsel i forhold til innlærte kunnskaper er derfor svært utslagsgivende for hvor mye opplæring som må gies. Opplæring har vært sett på som en stor utfordring K2IS i internasjonale fora tidligere. Ettersom yngre grupper som i større grad har vokst opp med Informasjonsteknologi også kommer inn i Forsvaret, vil opplæringsbehovet være mindre. En fordel man hadde tidligere var at forventningene ikke var alt for høye. Hvis opplæringskostnadene var høye, så var det fordi brukerne ikke hadde tidligere referanser eller kompetanse til å ta i bruk den nye teknologien. Nå har vi derimot kompetente og krevende brukere, med svært mange referanser til andre systemer. Gjenkjennbarheten vil derfor være svært viktig.³

K2IS er mer enn bare teknologi – det er heller et sosio-teknisk nettverk.

2.5.5 K2IS SOM DEL AV ANDRE NETTVERK

K2IS er som tidligere nevnt forsøkt avgrenset til utvalgte avdelinger i Hæren. Dette er av rent prosjekttekniske hensyn. Det er en verden utenfor, og hvordan skal K2IS samvirke med denne verden utenfor? Her er det to sentrale aspekter som bør nevnes. Det ene er interoperabilitet – samvirke –, det andre er sikkerhet – beskyttelse.

2.5.5.1 Krav til samvirke - interoperabilitet

K2IS skal i første omgang være et verktøy for Hæren i felt. Hæren samvirker med andre enheter så som luftstyrker, utenlandske enheter og overliggende kommandoer. Dette gjør at det er stilt en del krav til informasjonsutveksling. NATO har tradisjonelt gitt føringer på dette. Det finnes standarder som daterer seg tilbake til teleksteknologi, så vel som mer moderne EDIFACT-standarder. Den teknologiske utviklingen står ikke stille, og dette gjør at kravene til utvekslingsformater endrer seg. Et eksempel på dette er at det nå foreligger en standardisert datamodell for informasjonsutveksling med en tilhørende replikasjonsmekanisme, utviklet i MIP ^[MIP], et standardiseringsarbeid som pågår kontinuerlig.⁴

Norge har gjennom dette arbeidet forpliktet seg til å levere en MIP kapasitet i K2IS, det kan derfor hevdes at K2IS er sammenbundet med andre nettverk – og inngår derfor i et økosystem av nettverk.

2.5.5.2 Krav til sikkerhet

Målet for sikker databehandling er å ivareta konfidensialitet, integritet og tilgjengelighet i følge datasikkerhetsdirektivet [SikkLov] Med ny sikkerhetslov som nylig er iverksatt er dette blitt felles for all offentlig forvaltning i Norge.

³ I juli 01 uttaler Forsvarssjef Sigurd Frisvold at repetisjonsøvelser ikke lenger vil være vanlig i Forsvaret. Ut fra den uttalelsen han kom med til pressen kan det virke som at noe av opplæringen skal kunne foregå hjemmefra ved bruk av PC. Dette tydet i så fall på at likheten mellom Forsvarets bruk av Informasjonsteknologi og samfunnet for øvrig er blitt ganske stor. Fra Forsvarets ståsted styrker dette min påstand om at kompetente brukere vil stille krav om stor grad av gjenkjennbarhet når de skal ta i bruk K2IS.

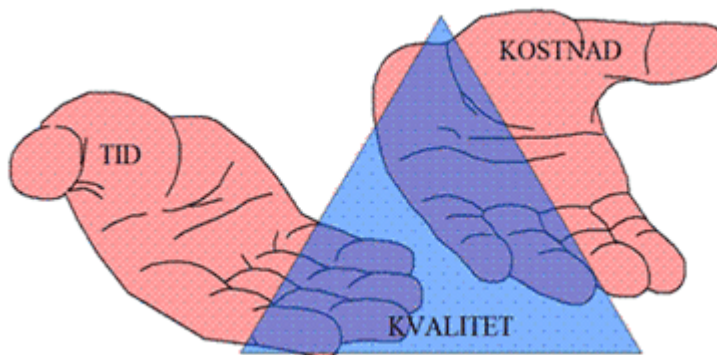
⁴ Norge ble i 1997 medlem i ATCCIS – Army Tactical Command and Control Information System, et standardiseringsorgan som primært hadde som mål å utvikle en felles informasjonsmodell for informasjonsutveksling. ATCCIS ble fra 2002 slått sammen med MIP. Mer informasjon om ATCCIS finnes på MIP sin hjemmeside, se dette [MIP].

Informasjonssystemene som skal benyttes i et taktisk miljø vil være utsatt for mange trusler. Uten å gå i detalj, kan vi bare fastslå at informasjon som er kritisk for en bruker, også vil være svært interessant for en potensiell fiende. Dette stiller krav til sikkerhet. Denne sikkerheten kan ivaretas på mange måter, eksempler kan være kryptering av informasjonskanaler, kontroll med brukertilgang osv. Sikkerhet er noe som blir anerkjent også i andre bransjer utenfor Forsvaret. Trusselbildet i for eksempel bank, børs og forsikring er kanskje vel så sammensatt som det Forsvaret ser for seg å møte. Tradisjonelt sett har man vært forsiktig med å gi innsyn i hvordan Forsvaret har løst slike utfordringer. I dag kan vi observere en holdningsendring på dette området. Denne endringen er på god vei allerede, dette skyldes ikke minst at tidligere Forsvaret Overkommando/Sikkerhetsstaben nå er Nasjonal Sikkerhetsmyndighet og derfor har tilsynsmyndighet for hele offentlig forvaltning. I forhold til stasjonære enheter som kan observeres over lang tid er det taktiske miljøet som K2IS skal operere i svært dynamisk. Dette gjør at kravene for det taktiske element heller enn å være strengere, faktisk kan settes lavere enn for mange andre tilsvarende løsninger. Dette har vi så langt ikke sett noen indikasjoner på internt, men hvis vi ser på hva som skjer hos enkelte av våre samarbeidende nasjoner, er det mulig å se tegn i denne retning. Et K2IS som skal samvirke med andre nasjoners tilsvarende systemer vil her påvirkes av de andre aktørers forhold til sikkerhet. Igjen er dette et eksempel på at K2IS ikke kan sees alene men som sammenbundet med andre nettverk – og inngår derfor i et økosystem av nettverk.

2.5.6 HELHETLIG REALISERING – HVORDAN TAKLE UTFORDRINGENE?

Vi har sett at utfordringene for K2IS er omfattende, og det kan hevdes at disse utfordringene går ut over det å anskaffe et ordinært kontorstøtteverktøy.

Hva så? Hvordan kan K2IS leveres til avtalt tid, til avtalt kvalitet og til avtalt pris? Dette er prosjektleders utfordring – ved realiseringen av K2IS som ved ethvert annet prosjekt, her illustrert i Figur 3.



Figur 4 Prosjektleders dilemma – avveining mellom tid, kostnad og kvalitet

Jeg velger å si at dette dilemma kan håndteres – og ønsker å bidra til dette gjennom denne oppgaven.

Jeg velger heller å si på samme måte som en gravemaskinfører kan styre grabben med to spaker- her gjerne tid og kostnad – så kan han grave ut det han ønsker – riktig kvalitet. Dette fordrer dog flere ting, forståelse for hvordan maskinen virker – hans verktøy – er det ene. En god forståelse er her nødvendig – og denne oppgaven vil presentere en slik helhetlig metode for å ta frem K2IS. Til en slik metode hører naturlig god opplæring – en erfaren gravemaskinfører bruker grabben som en forlengelse av sin egen arm. Jeg vil videre hevde at det er ikke prosjektleder som har den største utfordringen her, heller den organisasjon som han inngår i – altså hvordan kan organisasjonen utvikle K2IS over tid?

3 KRAV TIL METODE FOR SYSTEMUTVIKLING OG INTEROPERABILITET

Dette kapitlet tar utgangspunkt i forrige kapittel – anvendelsesområdet (Definert til den helhet hvor min besvarelse skal fungere – helheten av materiell, prosjekter, kultur, vaner og regler) – og kapitlet her skal da gi de krav som må stilles til en besvarelse av ”Hvordan kan store organisasjoner anskaffe informasjonssystemer som virker sammen over tid?”

Krav fremsetter jeg som et nødvendig kommunikasjonsmiddel mellom de fleste aktører - ingeniører, innkjøpere, ledere og kontrollere.

Dette kapitlet avsluttes med en oppsummering av krav til utviklingsprosess og hvordan oppnå interoperabilitet, som min besvarelse i de resterende kapitler skal forsøke å besvare.

3.1 KRAV – BEGREPSAVKLARINGER

I svært mange prosjekter står krav sentralt. Tradisjonelt har vi i alle fall i Forsvaret brukt mye tid på kravdokumenter. Disse kravdokumentene har forvaltningen etter analyse og tilbudsarbeider omarbeidet til en kravspesifikasjon. I en kontrakt med leverandøren er kravspesifikasjonen et av de viktigste dokumentene – betaling knytter seg til i hvilken grad kravene er blitt tilfredsstilt. I kapittel 0 vil jeg vise at kravdokumentet står som en svært sentral milepæl i en fossefallsmodell. I erkjennelsen av at ikke en kravspesifikasjon alene sikrer at kunden får det han egentlig ønsker, har nye metoder kommet med supplement til krav. Men selv for Rational Unified Process står fortsatt krav sentral. Hva er da krav? Jeg prøver i dette kapitlet å gi noen svar på dette. Jeg vil også vise noen av de krav som er satt i K2IS.

3.1.1 HVA ER KRAV?

For å gi et utgangspunkt til å forstå krav, velger jeg å starte med noen eksempler på krav.

Krav er beskrivelser av hvordan systemet skal oppføre seg, eller særlige egenskaper ved systemet. Krav kan også være bindinger til hvordan systemet skal utvikles.^[Sommer97]

Her følger noen eksempler på krav, listen er inspirert av Sommervilles' bok om kravhåndtering.

Eksempler på krav kan være:

- Editoren skal inneholde en kommando for stavekontroll og feilretting
- Systemet skal kunne Sikkerhetsevaluering til et gitt sikkerhetsnivå f. eks. HEMMELIG, jf. ^[SikkLov]
- Systemet skal sjekke sensorene 10 ganger i sekundet
- Systemet skal utvikles i Java

Disse kravene er eksempler på henholdsvis et brukerorientert krav, et overordnet systemkrav, et kvalitetskrav og et systemutviklingskrav.

Som vist i eksemplet over, kan krav fortelle oss en funksjonell egenskap ved systemet som skal lages – krav kan være knyttet til helheten – krav kan være føringer på systemutviklingen. Noen anbefaler at krav bare skal fortelle oss *hva* systemet skal gjøre, til forskjell fra hvordan det skal gjøres. Dette er en svært attraktiv ide, men dessverre er dette for enkelt i praksis.^[Sommer97] I det virkelige liv lever systemer sjelden alene. Andre systemer eksisterer allerede som det nye systemet skal samvirke med. For å få til dette, må det ofte legges krav til bruk av standarder og dette blir derfor begrensninger som utviklingen må ta hensyn til.

Krav inneholder derfor en samling av informasjon om problemområdet, utsagn om oppførsel og egenskaper samt føringer til design og utvikling.

Det er en kjensgjerning sier Sommerville at det er mange interessenter i forhold til et informasjonssystem. Svært mange er berørt ved innføringen av et system. Å samle inn alle deres krav er derfor en utfordring som han vektlegger sterkt. Det er altså forskjellige lesere av kravdokumentet, og dette vil avgjøre valg av formuleringer når kravene formuleres.

3.1.2 KATEGORISERING AV KRAV

Som tidligere nevnt har vi forskjellige typer krav. Her gjennomgår noen utbredte inndelinger.

3.1.2.1.1 Funksjonelle og ikke-funksjonelle krav

Disse begrepene har vi hatt mange diskusjoner på i vårt prosjekt. [Sommer97] gir en god definisjon som jeg bruker her.

Funksjonelle krav er krav som forteller hva systemet skal gjøre.

Ikke-funksjonelle krav setter begrensninger og rammer for hvordan disse kravene skal implementeres.

f. eks. kan et funksjonelt krav (relatert til eksemplet over på et overordnet krav) være at systemet skal tilby en tjeneste for at brukerne skal autentiseres, et ikke-funksjonelt krav vil da kunne være at autentisering skal gjennomføres på mindre enn 4 sekunder.

Dessverre sier Sommerville at denne delingen ikke alltid er så lett. Han beskriver et tilfelle knyttet til den samme situasjonen. Kravet om autentisering kunne da ha et tilleggskrav som sa at et spesielt signatursystem (som man vet bruker mindre enn 4 sekunder) skulle brukes. Er dette siste kravet et funksjonelt krav eller et ikke-funksjonelt krav? Sommerville gir ikke svaret. Sommerville sier at ofte brytes overordnede ikke-funksjonelle krav ned i funksjonelle systemkrav.

3.1.2.1.2 Funksjonelle krav, kvalitetskrav og verifikasjonskrav.

I utviklingen av K2IS spilte Våpenskolen for Hærens samband (VSHSB) en sentral rolle. VSHSB drev i en årrekke med blant annet virksomhetsmodellering. VSHSB gjennomførte utledning av krav etter virksomhetsanalyse. Det vises til rapporten ^[EvalPR99] utarbeidet av SINTEF Tele og Data som går nærmere inn på det omfattende og svært krevende arbeid VSHSB utførte innenfor K2IS.

Et at de resultatene som VSHSB produserte var krav til teknologi som skal understøtte virksomheten. Dette er krav til sambands- og informasjonssystemene som skal brukes innenfor K2IS i Hæren. VSHSB har gjort en annen deling som nok er vel så formålstjenlig som delingen funksjonelle/ikke-funksjonelle. Her gjengis kort denne deling ^[GJFPL99];

Funksjonelle krav	Beskriver HVA systemet gjør, ikke hvor godt eller kostnader forbundet med det.
Kvalitetskrav	Angir HVOR GODT eller med HVILKEN KVALITET en funksjon utføres
VERIFIKASJONSKRAV	Angir HVORDAN krav skal verifiseres, og hvilke AKSEPTANSEKRITERIER som legges til grunn for godkjenning.

Tabell 1 Inndeling av krav

Det finnes altså flere typer krav. Hensikten med en inndeling kan for eksempel være å rette kravene mot å fremme dialog i en målgruppe – mellom brukeren og applikasjonsdesigneren, mellom prosjektledelsen og brukerrepresentantene eller mellom system engineering hos leverandør og eget testpersonell. Når jeg skal kategorisere krav velger jeg å bruke denne siste inndelingen.

3.2 KRAV TIL EN METODE

Jeg har spørsmålet: ” Hvordan kan store organisasjoner anskaffe informasjonssystemer som virker sammen over tid?”

Stikkord her er organisasjoner og gjerne da store organisasjoners krav til anskaffelse av informasjonssystemer. Videre er stikkord ”virker sammen” og enda mer ”over tid”.

Det siste først – virker sammen over tid. Jeg velger å kalle et slikt samvirke for interoperabilitet, og kommer nærmere inn på dette i neste kapittel.

Virker sammen over tid sier noe om å håndtere endringer som måtte skje, dette kan knyttes både til systemutvikling og interoperabilitet. Jeg velger å beskrive egenskapen over tid som kontinuerlig. Det vil si at beskrivelsen må tåle endringer. Dette kan innebære alt fra endringer i organisasjon, personell, arbeidsprosesser eller endringer i teknologi.

I systemutvikling legger jeg fremskaffelse eller anskaffelse av informasjonssystemer. Jeg får da to hoverkategorier - interoperabilitet og systemutvikling.

Det viktigste ordet i oppgavens formulering er hvordan. ”Hvordan” fordrer at jeg formulerer noe som beskriver en fremgangsmåte som kan brukes – dette kaller vi en metode. Slik vil svaret på spørsmålet i oppgaven være en metode, der kravene stilles til denne metoden med hovedvekt på interoperabilitet og systemutvikling og hvordan dette håndteres kontinuerlig.

3.3 HÅNDTERING AV INTEROPERABILITET

Kravet til interoperabilitet er i dag uttalt i mange sammenhenger. Hva er interoperabilitet? Hva mener vi med at noe skal være interoperabelt? Kan interoperabelt måles? – kan vi snakke om grader av interoperabilitet? Dette forsøker jeg å si noe innledningsvis om i det som nå følger.

3.3.1 INTEROPERABILITET - EN INTRODUKSJON

3.3.1.1 Begrepet interoperabilitet

Begrepet interoperabilitet er ikke nytt for det norske forsvar. Begrepet har lenge vært i bruk innenfor NATO, dette kan sees blant annet i ^[PTG,98] Et ytterligere fokus på interoperabilitet som samvirke mellom nasjoner og styrkekomponenter har kommet etter NATO toppmøtet i Praha ^[Praha02], hvor starten på den siste utvidelsen av organisasjonen ble iverskatt. Videre har ikke minst de senere års fokus på nettverksbasert forsvar satt interoperabilitet i Fokus. Her kan nevnes at i den siste Stortingsproposisjon for Forsvaret for inneværende periode 2005-08^[St.pr-42] er ordet ”interoperabilitet” benyttet 11 ganger svært ofte i samme betydning som ordet ”samvirke” som er benyttet 17 ganger. Nettverksbasert forsvar har lenge vært termen brukt her i Norge, men det kan være som dette er i ferd med å bli presisert som nettverket som en styrkemultiplikator. Dette er i så fall langt på vei i tråd med det NATO har pekt ut med sin strategi for NATO Network Enabled Capabilities ^[NNEC]. En nødvendig forutsetning for interoperabilitet vil i mange sammenhenger være avhengig av en god informasjonsutveksling, også gjerne kalt tekniske interoperabilitet. Informasjonsutvekslingen må møte mange ulike krav for å være troverdig. Denne avhengigheten gjelder mellom samvirkende nasjonene så vel som innefor den enkelt nasjon og de nasjonale forsvarsgrener. Disse kravene kan være så vel semantisk preservasjon som tidsaspekter og ulike grader av lesbarhet. For å øke effektiviteten i organisasjonen generelt, benyttes automatisk støtte for informasjonsbehandling og

informasjonsoverføring i organisasjonen. Denne automatiseringen fordrer økt kvalitet i spesifikasjonen av informasjonen. Ved automatisering er ikke menneskelig tolkning tilstrekkelig for å oppnå den kvalitet som er krevd. I denne hovedoppgaven er det brukerperspektivet som beskrives. Interoperabilitet mellom kommunikasjonsenheter, valg av standarder på kommunikasjonsbærere sees på som virkemidler for å oppnå interoperabilitet. Nettperspektivet brukes derfor bare for å belyse brukerperspektivet der dette anses nyttig. Dette er tilsvarende som i [PTG,98]

3.3.1.2 Interoperabilitet – mange fasetter

Interoperabilitet - evnen organisasjoner, systemer, informasjon, teknologi og infrastrukturer har til å operere sammen⁵. Begrepet favner altså ikke bare teknologiaspekter, ei heller bare organisatoriske aspekter.

Begrepet interoperabilitet brukes mye i militære fora. Grunnen til at begrepet er så utbredt er nok at dette ordet har hatt en anvendelse lenge før det var aktuelt å snakke om Edb-baserte informasjonssystemer.

3.3.1.2.1 Interoperabilitet – et operativt krav

Fra et operativt ståsted har man lenge hatt et forhold til begrepet. Fra norsk ståsted er dette gjort gjeldende som krav til våre styrker som skal delta i internasjonale operasjoner^[ST-M 38] og^[ST-P 42]

3.3.1.2.2 Interoperabilitet – et internasjonalt krav

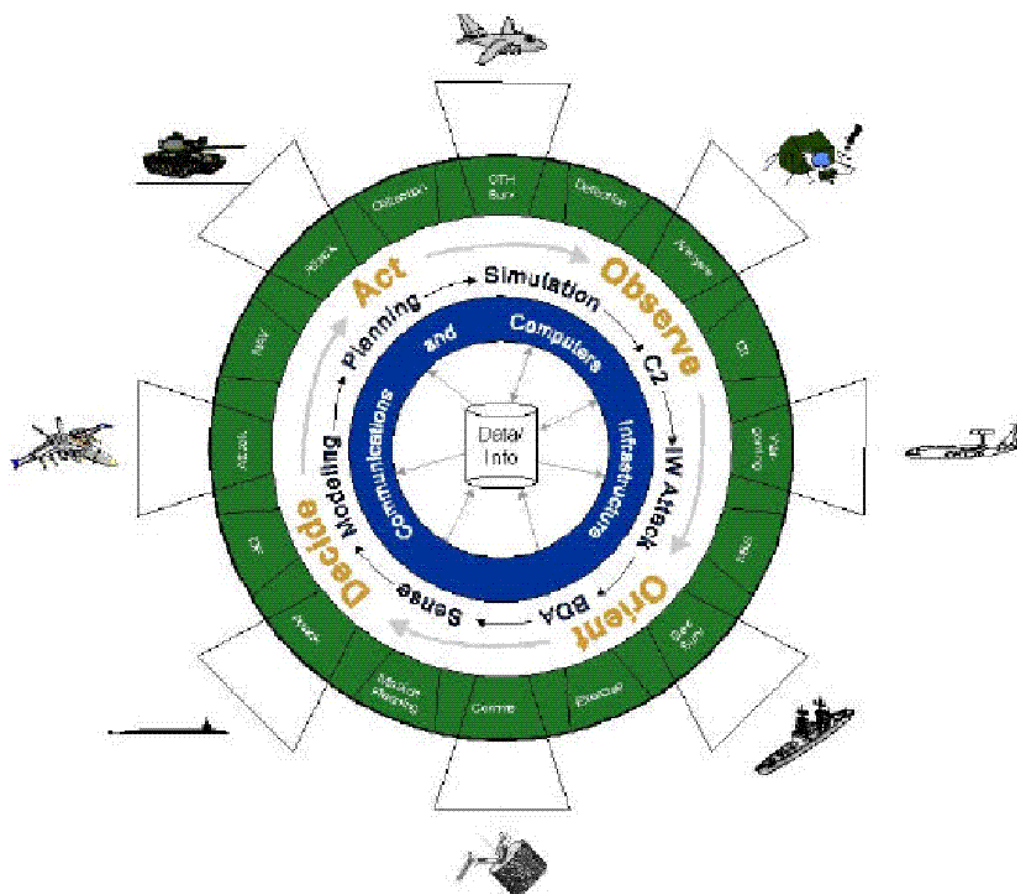
Forsvaret i USA har lenge hatt begrepet på agendaen. Jeg deltok selv på en konferanse for noen år tilbake i USA, Command & Control Research Symposium-1999^[CCRP-99] hvor et av 6 temaer var interoperabilitet.

I innledningsforedraget på denne sesjonen skisserer Dr Curts kravet til interoperabilitet sett fra et operativt ståsted. Figur 4 Kravet til Interoperabilitet er hans illustrasjon av dette.

⁵ Interoperabilitet - Interoperability – the ability of organisations, systems, information, technologies and infrastructures to operate together, DISGIS: An Interoperability Framework for GIS – Using the ISO/TC 211, Model-based Approach, <http://folk.uio.no/roygr/GSDI-2000.pdf>

^[ST-M 38] Stortingsmelding 38. 98-99, Tilpasning av Forsvaret til deltagelse i internasjonale operasjoner, Kapittel 3.3 Prinsipper for operativ flernasjonaltitet, " Den første forutsetningen for flernasjonaltitet er at de angjeldende styrker i tilstrekkelig grad er i stand til å operere effektivt sammen for å løse de aktuelle oppgavene. Dette innebærer at styrkene må ha standardiserte - eller i det minste kompatible - prosedyrer for blant annet kommando og kontroll"

^[ST-P 42] [St.pr-42] Det vil være ufordelaktig, og kunne svekke norsk sikkerhet på sikt, dersom norske militære styrker ikke er i stand til å virke sammen med allierte styrker. Interoperabilitet med allierte er således både motivert ut i fra et nasjonalt og et internasjonalt perspektiv.



Figur 5 Kravet til Interoperabilitet

Her illustreres at våpensystemer på bakken (stridsvogner)- på sjøen (krigsskip)- og i luften med (kampfly)-, i tillegg til å virke hver for seg også skal virke i forhold til hverandre. Dette er et ambisiøst krav og kan lett assosieres med kravene til kongen som ba om et skip som gikk både til land, til vanns og i luften med, i eventyret om Askeladden og de gode hjelperne. Visjonen er at de enkelte delelementene satt sammen i en enhet skal forsterke hverandre. Dette er mulig å forstå, og bør også kunne være et rimelig krav.

3.3.1.2.3 Interoperabilitet – et krav til organisasjonen

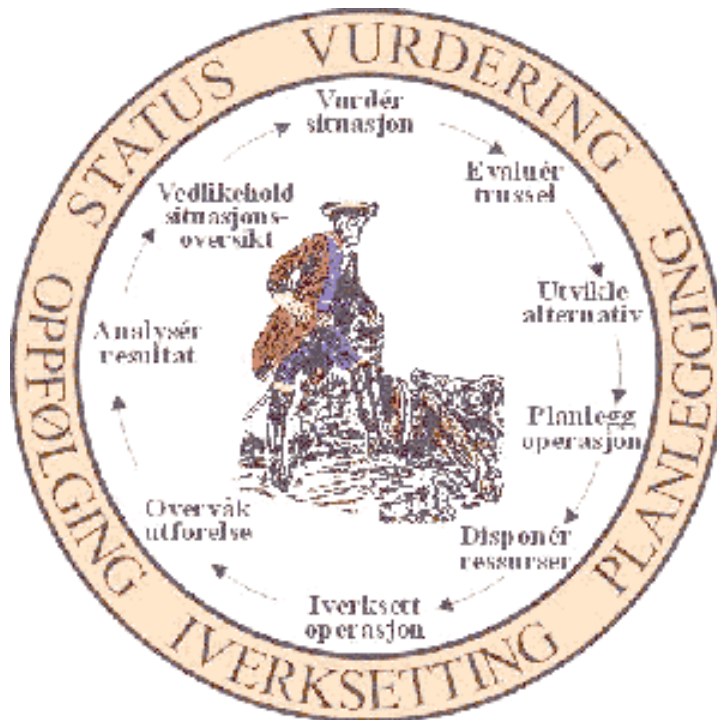
Videre skal det i figuren være mulig å se at det er et sett av funksjoner som utøves mot våpensystemene, disse funksjonene igjen ledes gjennom ledelsesfunksjoner. Kommando og kontroll eller K2 er ett sett av ledelsesfunksjoner – disse er illustrert i Figur 6 Kommando og kontrollhullet.

Dette er en allmenngyldig prosess som den operative leder bruker for å utøve kommando og kontroll ⁶.

⁶ Samlede aktiviteter som utøves i en Operativ kontekst.

Kommando: planlegging, styring, ressursfordeling. Kontroll: monitorering og eventuell iverksettelse av korrektive tiltak.

Note: Termen Kontroll er her benyttet som en direkte oversettelse av det amerikanske 'Control' som generelt betyr styring eller regulering. Den norske termen 'Kontroll' dekker egentlig bare monitoreringsdelen av 'Control'. [Termbank, K2IS Blokk 1]



Figur 6 Kommando og kontrollhjulet

I Norsk sammenheng sier vi at denne operative funksjonen forholder seg til en kontekst, Kommando og Kontroll system ^[K2S]. Denne funksjonen skal virke ved å lede andre feltfunksjoner, manøverfunksjoner, taktisk støtte, logistikk, samband og så videre. Interoperabilitet er altså et krav til samvirke i organisasjonen.

3.3.1.2.4 Interoperabilitet – et krav til tekniske system

Som forsøkt illustrert her er det altså en relasjon mellom funksjoner i virksomheten og funksjoner i et teknisk system.

Hva betyr så dette for utviklingen av Informasjonssystemer? Som Dr Curts ved [CCRP-99] så vel formulerte det, ”dessverre har evnen til å generere informasjon i dag langt overskredet evnen til å kontrollere og bruke informasjonen effektivt”. Fra kravene til interoperabilitet mellom operative elementer til hva dette betyr for informasjonssystemer er det en relasjon, men denne relasjonen virker noe diffus. Så langt kan vi si at relasjonen er tosidig - det har lenge vært stilt krav til systemene fra de operative, men det vil også kunne være slik at systemene stiller krav tilbake til det operative miljøet. Dette flerperspektivet skal jeg komme tilbake til flere ganger.

3.3.2 AGENDA FOR FREMTIDENS FORSVAR

Våren 1999 på Ministermøtet i Washington DC ble Defence Capability Initiative oppdatert. DCI ^[DCI-99] er først og fremst kapasiteter og kvaliteter på operasjonelt nivå (capabilities) og ikke teknologi- og spesifikke våpen- systemtyper. ^[FFI-3429] I bakgrunnsdokumentasjonen for statsoverhodenes møte, er det understreket at fremtidens utfordringer og oppgaver for alliansen vil bli svært forskjellig fra det vi planla for under Den kalde krigen. Det siste tiåret har demonstrert dette med all ønskelig tydelighet,

^[K2S] Doktriner, metoder, prosedyrer, personell, utstyr og fasiliteter organisert for utførelse av K2. [Termbank, K2IS Blokk 1]

men medlemslandene har i begrenset og varierende grad tatt konsekvensene av dette og endret sine forsvar. Uansett hvilken vinkling man velger å anlegg på DCI, er prosessen helt avhengig av konkrete målsettinger og en tett oppfølging. Målsettingene er formulert med utgangspunkt i det erkjente svakheter i dagens situasjon. På Washington toppmøtet presiseres det at behovene for endring og forbedring primært er knyttet til reaksjonsevne, utholdenhet, kommando/kontroll/informasjon og beskyttelse. Dessuten er interoperabilitet en gjennomgående rød tråd i EWG-dokumentet som NATOs Generalsekretær la frem. Det understrekes at evne til å operere sammen er en avgjørende forutsetning for effektiviteten i alliansens operasjoner. Multinasjonale styrkeformasjoner vil bli fremtidens løsning, og denne multinasjonaliteten vil måtte gå ned til relativt lave organisasjonsnivåer. DCI lister fem hovedområder

- Deployability and Mobility
- Sustainability and Logistics
- Effective Engagement
- Survivability of Forces and Infrastructure
- NATO Command and Control and Information Systems.

Av dette fremkommer det et spesifikt krav til interoperabilitet under utholdenhets- og forsyningsmessige krav nr 5a "NATOs nasjoner må forsterke interoperabilitet gjennom økt standardisering av materiell og prosedyrer og ved realisering av felles standarder.⁷

Her snakkes det om både standardisering av materiell og prosedyrer. Kravet til materiellet er krav direkte til teknisk utstyr og deres funksjoner. Det er også et krav til standardisering av materiell og dette er valg som styrer utviklingen av materiellet – et systemutviklingskrav.

Det er altså et kombinert krav til teknologi og menneskelig adferd.

3.3.3 METODE FOR Å HÅNDTERE INTEROPERABILITET

Jeg må altså beskrive hvordan interoperabilitet håndteres kontinuerlig. Jeg har sagt at interoperabilitet er både organisasjon, prosess og teknologi. Dette innebærer at jeg må ha en metode som kan beskrive interoperabilitet i en organisasjon både mellom organisatoriske element og systemelement. Videre må denne metoden håndtere endring kontinuerlig.

3.4 HÅNDTERING AV SYSTEMUTVIKLINGEN

Denne oppgaven tar for seg utviklingen av et system i en større organisasjon – Hæren. Systemet som skal lages, K2IS blir dog utviklet helt uavhengig av denne oppgaven. Det oppgaven forsøker å sette fokus på er hvordan bør et slikt system utvikles. Hvilke krav er det til en eventuell systemutviklingsmodell eller metodikk som skal følges? Dette kapitlet forsøker å sette krav til en slik systemutvikling og hva en foreslått metode bør inneholde.

Ut over det at systemer skal "snakke sammen" – les kravet til interoperabilitet, skal det også beskrives hvorfor vi har systemene, hva de kan brukes til, hvordan dette henger sammen med viktige deler av virksomheten, hvordan lover og regler influerer systemutvikling/fremskaffelse/anskaffelse og hvordan dette skal håndteres kontinuerlig. Mitt case er her Forsvaret og K2IS i Hæren spesielt – så her benyttes denne konkrete settingen for kravene.

⁷ DCI, Sustainability and Logistics (SL), SL 5a: NATO Nations should enhance interoperability through increased standardization of materiel and procedures and the implementation of common standards.

3.4.1 KRAV TIL SYSTEMET

Tekniske systemer som K2IS har sin berettigelse ved at de er til nytte for de som bruker systemet. Systemene må derfor innfri brukernes forventninger.

En erkjennelse som også ligger i oppgaven er at et ”ferdig system” ikke vil være nettopp det – ”ferdig” svært lenge. Endringer i omgivelser, eller endringer av behov eller teknologi tilsier at systemet også må håndtere dette kontinuerlig.

Systemene skal være enkle å oppdatere ettersom kravene fra omgivelsene endrer seg.

En metode for systemutvikling må håndtere disse kravene til systemutviklingen.

3.4.2 KRAV TIL SYSTEMUTVIKLINGEN

De krav som er stilt til systemet må systemutviklingen ta inn over seg.

En metode for systemutvikling må derfor håndtere forventninger, funksjonelle krav og kvalitet. Det er systemutviklingen som skal sikre at brukernes forventninger blir innfridd.

Videre er det en del andre krav som alltid vil være knyttet til utvikling og endring.

Dette er krav knyttet til finansiering samt ledelse både av risiko og arbeidsprosesser.

Metoden må derfor håndtere usikkerhet, kommunisere de risikoelementer som kan identifiseres og de tiltak som iverksettes for å håndtere identifisert risiko. Systemutviklingen skal bidra til at opplevd usikkerhet for de som er involvert avtar i utviklingsløpet.

Metoden må videre sikre at systemene kan utvikles med lavest mulige kostnader. Metoden må samtidig håndtere endringer i systemutviklingsorganisasjonen slik at endringer i personell, oppbygging av kompetanse etc. kan håndteres på en forsvarlig måte.

Systemutvikling i Forsvaret er for en stor del drevet frem gjennom anskaffelser. Systemutviklingen vil derfor påvirkes sterkt av regler for anskaffelse.

3.4.3 ANSKAFFELSER OG SYSTEMUTVIKLING

Anskaffelsesmetode skal sikre at anskaffelsen er i tråd med Lover og regler, herunder gjeldende regler for anskaffelser i forsvaret.^[ARF]

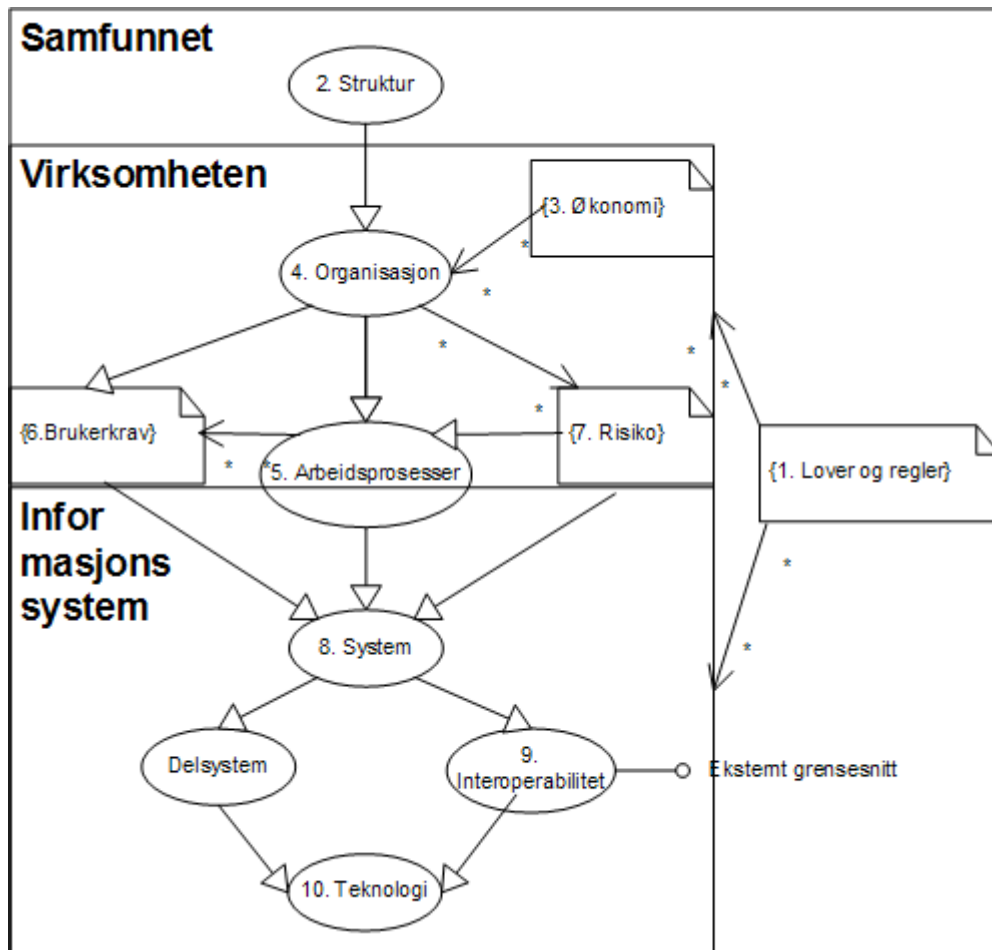
Anskaffelsesmetoden skal sikre at systemutviklingen leverer et system som passer inn i den helhet som Forsvaret har vedtatt.

Anskaffelsesmetoden skal sikre at systemutviklingen kan gjennomføres som en åpen handling mellom kunde og leverandør for dermed å sikre at kundens forventninger blir innfridd.

Anskaffelsesmetoden skal sikre at Forsvaret får den beste støtte for å redusere risiko i gjennomføring av sin anskaffelse.

3.5 OPPSUMMERING

En illustrasjon av de elementer som en metode for systemutvikling og interoperabilitet er gjort i følgende modell, se Figur 7:



Figur 7 De 10 elementene

Kravene er oppsummert slik:

	Håndteres	Hvordan håndtere endring:
Lover og regler	Ja	N/A
Struktur	På en åpen måte	Synliggjøre konsekvenser
Økonomi	Alternative løsninger	Synliggjøre konsekvenser
Organisasjon	På en åpen måte	Holde à jour (Enkelt å oppdatere)
Arbeidsprosesser	På en åpen måte	Holde à jour (Enkelt å oppdatere)
Brukerkrav	På en åpen måte	Synliggjøre konsekvenser
Risiko	På en åpen måte	Synliggjøre konsekvenser og tiltak
Systemer	På en åpen måte	Holde à jour (Enkelt å oppdatere)
Interoperabilitet	På en åpen måte	Holde à jour (Enkelt å oppdatere)
Teknologi	På en åpen måte	Synliggjøre konsekvenser

Tabell 2 Krav til oppgaven

4 EKSISTERENDE LØSNINGER

Min oppgave tar for seg hvordan store organisasjoner kan anskaffe systemer som virker sammen over tid. Når jeg da skal se på hva som eksisterer innenfor dette området, har jeg plukket ut det jeg har funnet sett som vesentlig innenfor systemutvikling og interoperabilitet. Jeg tar utgangspunkt i disse to emnene, for her finnes det mye stoff. For å supplere på systemutviklingsteori, har jeg tillagt et kapittel om rammeverk. Dette er nytt stoff i forhold til mine fag fra IFI, men vesentlig for denne oppgaven.

Videre tar jeg for meg informasjonsinfrastruktur, som omhandler hvordan standardisering og utvikling av en infrastruktur konvergerer og divergerer.

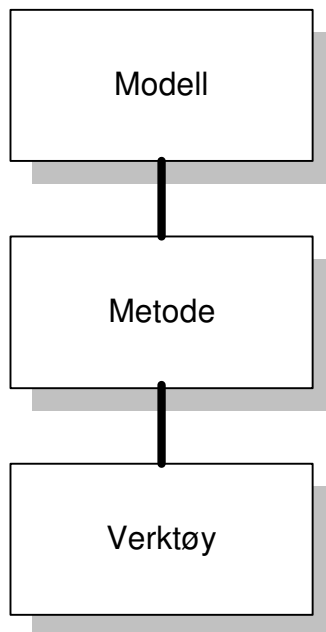
Avslutningsvis tar jeg for meg elementer som fremkommer i tillegg til ren systemutvikling, når vi snakker om store organisasjoner – IT-strategi og anskaffelse sett i forhold til intern utvikling.

4.1 SYSTEMUTVIKLING

4.1.1 BEGREPSAPPARAT

I kurset IN 364 Systemarbeid: Teori og modeller ved Institutt for Informatikk, ble det skissert et hierarki mellom modeller, metoder og verktøy. Modellen ligger over og gir en ramme, en øvre filosofi eller en måte å tenke på. Metode er de måter man jobber på for å nå de mål man setter seg. Verktøy er det laveste nivået som inneholder de virkemidler man benytter når man jobber som metoden beskriver. Tanken bak hierarkiet er at det muliggjør endringer på ett nivå uten at det skal være nødvendig å gjøre endringer på nivået over. Kapittel 4.1.3 Utviklingsmodeller omtaler det øverste nivået, modellnivået, og omtaler tre slike modeller. Ideen med modellene er at de er noe som ikke endrer seg så ofte. Disse ligger da som en overbygging som kan leve over tid, og som da en valgt metode forsøker å leve opp til. Metodene er da mer eller mindre spesifikke etter hvilken modell man vil følge. Uten å kunne føre noe bevis for det påstår jeg at siden fossefallsmodellen har vært lengst på markedet, finnes det også flest metoder som bygger oppunder denne. Metodene utvikles med tiden og for å tilfredsstille nye behov og ønsker, jeg gir eksempler på noen metoder i det følgende. For å støtte metodene kommer det stadig nye verktøy på banen. Caseverktøy er i dag en naturlig del av ethvert systemutviklingsprosjekt. Verktøy er ofte teknologiavhengig. Med en utvikling som så langt har fulgt Moore's ^[Moore's] lov endrer maskiner og dertil hørende programvare seg svært raskt. Dette gjør at verktøysiden av dette hierarkiet er svært dynamisk. En illustrasjon av de tre nivåene er forsøkt gjort i Figur 8 Modell - Metode - Verktøy, et hierarki.

^[Moore's] Moore's lov; En lov basert på empiriske studier utviklet og presentert Intels Gordon Moore. Loven forutser at industrien som lager integrerte kretser (IC'er) vil kunne doble antallet transistorer på en silisiumbrikke hvert annet år. (senere tolkninger sier i dag 18 måneder) De fleste utviklingssykluser i den elektroniske industri baserer seg på Moores lov. Se for øvrig http://en.wikipedia.org/wiki/Moore%27s_Law



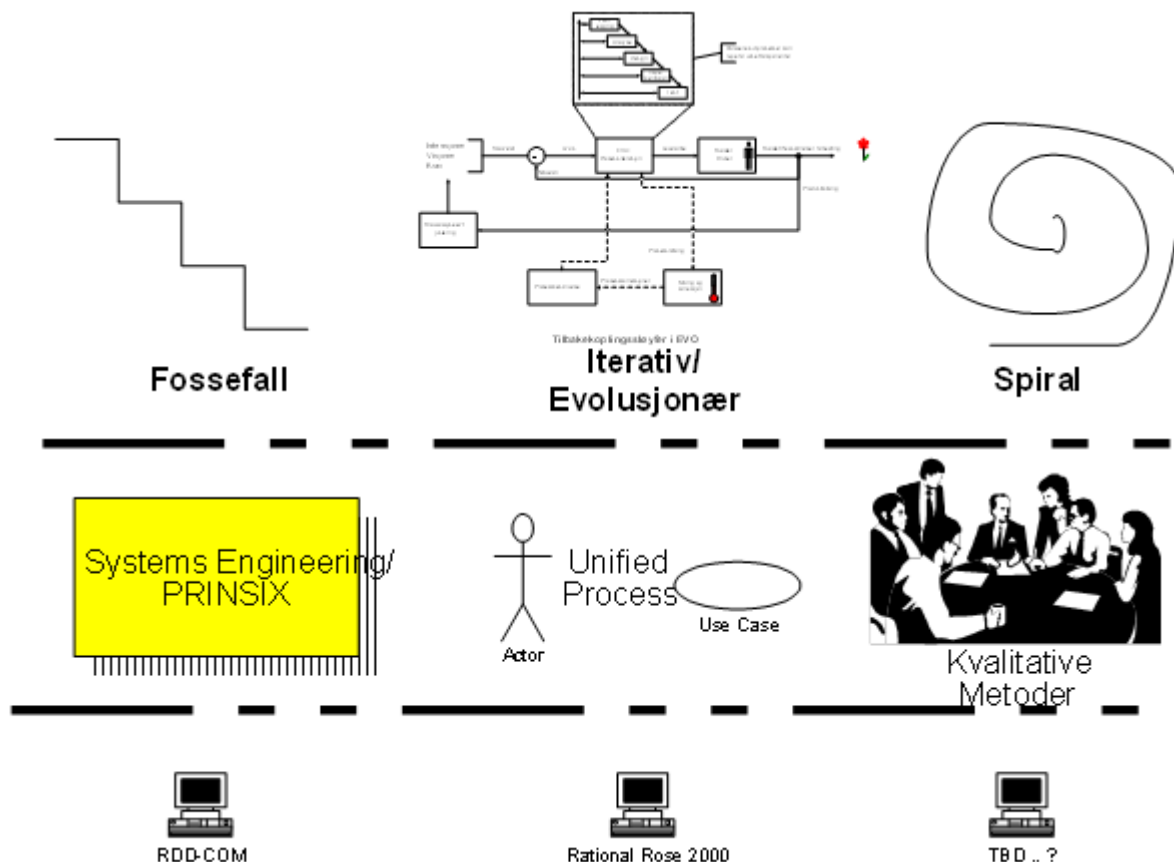
Figur 8 Modell - Metode - Verktøy, et hierarki

Det skal være mulig å velge forskjellige typer verktøy innen for en utviklingsmetode. Det skal også være mulig å anvende flere enn en metode innenfor en modell.

4.1.2 EN TREDELT TILNÆRMING

Begrepsapparatet som blir benyttet i innenfor systemutvikling baserer seg på tre hovedelementer – modell – metode og verktøy. Kapittel 4.1.3 tar for seg tre slike modeller. Ideen med modellene er at de er noe som ikke endrer seg så ofte. Disse ligger da som en overbygging som kan leve over tid, og som da en valgt metode forsøker å leve opp til. Metodene er da mer eller mindre spesifikke etter hvilken modell man vil følge. Uten å kunne føre noe bevis for det påstår jeg at siden fossefallsmodellen har vært lengst på markedet, finnes det også flest metoder som bygger oppunder denne. Metodene utvikles med tiden og for å tilfredsstille nye behov og ønsker, jeg gir eksempler på noen metoder i kapittel 4.1.4. For å støtte metodene kommer det stadig nye verktøy på banen. Caseverktøy er i dag en naturlig del av ethvert systemutviklingsprosjekt. Verktøy er ofte teknologiavhengig. Med en utvikling som så langt har fulgt Moore's lov endrer maskiner og dertil hørende programvare seg svært raskt. Dette gjør at verktøysiden av dette hierarkiet er svært dynamisk. Jeg omtaler kort noen verktøy i kapittel 4.1.5.

En illustrasjon av de tre nivåene med noe av innholdet som blir presentert er gjort i Figur 9 Modell - Metode – Verktøy.



Figur 9 Modell - Metode – Verktøy

4.1.3 UTVIKLINGSMODELLER

I den etter hvert klassiske artikkelen til Barry Boehm^[Boehm], hvor spiralmodellen introduseres, presenteres denne som en av tre hovedmodeller - Vannfall, Evolusjonær og Spiralmodell.

4.1.3.1 Vannfallsmodellen

Vannfallsmodellen er en vel utbredt modell. Etter hver fase har modellen en milepæl med leveranse av et produkt. I tidlige faser er dette dokumenter, alle faser avsluttes med leveranse av dokumenter, modellen er derfor å være sterkt dokumentdrevet. Modellen egner seg godt for å måle fremdrift i den grad en leveranse av et produkt er et fremskritt. Begrepet "Vannfall" er nok fremmet av de som ønsker å kritisere metoden. Blant kritikerne sies det at det er "som et vannfall, når fasen er avsluttet - når milepælen er passert går man ikke tilbake, på samme måten som at vannet ikke renner oppover i et vannfall. Dette oppfattes derfor som en veldig rigid modell. Modellen er likevel ikke på noen måte død. Svært mange prosjektstyringsmodeller, så som Forsvarets prosjektstyringsmodell, PRINSIX er en vannfallsmodell. Modellen er også sterk innenfor tradisjonelle industrier og bygg og anlegg.

4.1.3.2 Evolusjonær modell

For å løse opp på den sterke formalismen i vannfallsmodellen ble en mer evolusjonær modell lansert. Tanken er at det er en kontinuerlig endring, denne endringen består av små skritt som på sikt endrer

helheten. Et biosystem vil ha tilsvarende egenskaper, der en endring som blir innført også vil kunne få innvirkning på andre enheter over tid. Modellen blir kritisert for å være lite kontrollerbar.

4.1.3.3 Spiralmodellen

Som en syntese over disse lanserte derfor Boehm spiralmodellen[Boehm]. Denne modellen skal i stor grad ta inn i seg de gode egenskapene fra begge de to andre modellene. Risiko er styringsparameteren. Modellen er syklisk, man går gjennom de samme fasene flere ganger, og gjør de nødvendige tiltak for å redusere risiko. Under kurset på Blindern fikk jeg ikke noen konkrete eksempler på at modellen var anvendt i noen særlig grad. Dette vekket min nysgjerrighet.

4.1.4 UTVIKLINGSMETODER

Det finnes mange metoder på markedet. En hver større organisasjon som driver utvikling av software gjør dette etter en metode. En trend i tiden er at svært mange søker mot standardmetoder. Dette kan skyldes at man ikke ønsker å ha egne metoder, at standardmetoden(e) er i ferd med å bli godt beskrevet og dermed svært anvendbare, eller muligens en kombinasjon av dette.

4.1.4.1 Systems Engineering metode

Tilbake i 1990 ble en forening etablert i USA for systems engineering, “the National Council on Systems Engineering” (NCOSE). NCOSE hadde som formål behovet for forbedringer innenfor systems engineering metodikk og undervisning. Dette førte videre til en økt involvering av systemingeniører utenfor USA, og dette førte til at organisasjonen endret navn til the International Council on Systems Engineering (INCOSE) i 1995.

US DOD utga i 1992 MIL-STD-499B som en omfattende revisjon av den tidligere MIL-STD-499A, Engineering Management. Navnet skulle være “Systems Engineering”. Det viste seg å være vanskelig å oppnå konsensus blant industrideltakerne om hva som burde inngå i en slik standard.

I 1994, kunngjorde DoD at MIL-STD-499B ikke ville bli utgitt som en egen militær standard. Dette førte til at Electronic Industries Alliance (EIA) satte ned en arbeidsgruppe med bred deltakelse. Resultatet var i 1994 en kommersialisert versjon av MIL-STD-499B som EIA Interim Standard (IS) 632, [EIA/IS 632 1994]. Målet var etter hvert som flere industriparters bidrag ble inkludert skulle dette ende ut som en oppdatert versjon kalt EIA 632. I parallell utga IEEE i 1995 en prøveversjon av standard IEEE 1220-1994. IEEE 1220-1994 skulle slås sammen med EIA/IS 632 til en American National Standards Institute (ANSI) standard for systems engineering, som da skulle bli publisert av EIA, IEEE, and INCOSE i samarbeid.

IEEE-1220^[IEEE1220] er IEEE's Standard for anvendelse og ledelse av Systems Engineering prosessen. I prosessbeskrivelsen inngår arbeidsoppgaver i flere disipliner som er nødvendig gjennom hele livssyklusen til et system fra behov, krav og begrensninger til en endelig systemløsning.

Metoden er ikke undervist som en del av de kursene jeg kjenner til ved IFI. IEEE 1220 kan tradisjonelt oppfattes som å sterkt underbygge en vannfallsmodell, og er svært dokumentorientert. Metoden kommer fra tradisjonell system engineering og har ikke opprinnelig noen spesielle egenskaper som knytter den til software engineering. Den ivaretar dog en helhet ved å ta for seg både software og hardware. Det har vist seg at noen av våre samarbeidsparter har denne metoden som ballast. Metoden er dokumentert, godt innarbeidet og til dels har den hatt svært god støtte av verktøy.

Våpenskolen for Hærens Samband gjennomførte tidligere sine aktiviteter innefor systemutvikling med støtte i en System Engineering Management Plan [GJFPL99] basert på IEEE-1220.

4.1.4.2 Unified Process

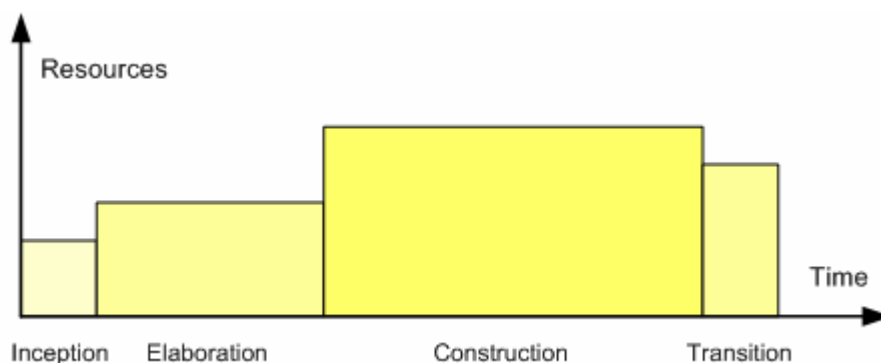
Unified Process^[UP] er en iterativ og inkrementell metode for software utviklingsprosessen. Den best kjente og også den best dokumenterte raffinering av Unified Process er Rational Unified Process, også kjent som RUP. RUP ble utarbeidet av Booch, Rumbaugh og Jacobson og bakgrunnen for etableringen av firmaet Rational, siden 2003 en del av IBM.

4.1.4.2.1 Hovedelementer

RUP bygger på prinsippene om å utvikle software iterativt, håndtere krav, benytte en komponentbasert arkitektur, benytte visuell modellering samt verifisere software kvalitet ved å kontrollere endringer underveis.

4.1.4.2.2 Metodens faser og iterasjoner

Prosessen baserer seg i kort på å fordele de definerte arbeidsstegene i metoden i en delvis ordnet sekvens av handlinger. RUP livssyklus organiserer arbeidet i fire faser, inception, elaboration, construction og transition. Figur 10 viser en profil med fokus på innsats og omfang for et typisk prosjekt



Figur 10 Fasene i RUP, visert omfanget for et typisk prosjekt

I Inception fasen vektlegges anvendelse, virksomhetsmodell, suksesskriterier, og antagelse om fremtidig fortjeneste. I tillegg til en virksomhetsmodell, utarbeides en første anvendelsesmodell, prosjektplan, plan for risikohåndtering og prosjektbeskrivelse. Når dette er fullført sjekkes prosjektet mot et sett med kriterier. Eksempler her er aksept av prosjektets målbilde og kostnadsestimater, tiltro til kostnadsestimater samt dybde og bredde til prototyper som er utviklet. Hvis prosjektet ikke passerer milepælen, Lifecycle Objective Milestone, termineres prosjektet eller denne fasen repeteres til prosjektet er bedre tilpasset kriteriene.

I elaboration fasen begynner prosjektet å ta form. I denne fasen gjennomføres analyse av problemområder og den grunnleggende arkitekturen kommer på plass. Denne fasen avsluttes med Lifecycle Architecture Milestone hvis en serie med kriterier er møtt. Eksempler her er en use-case modell hvor use-cases og aktører er identifisert og bør være 80 % komplett. Videre skal software arkitekturen være beskrevet som del av en systemutviklingsprosess og en eksekverbar arkitektur som utfører de arkitektonisk viktigste use-cases skal være plass. Hvis prosjektet ikke passerer denne fasen

er det fortsatt tid for at det kan kanselleres eller redesignet. Etter denne fasen går prosjektet inn i en høyrisiko fase hvor endringer blir vanskeligere å utføre og får større konsekvenser.

I Construction fasen vektlegges utvikling av komponenter og andre deler av systemet. I denne fasen blir det meste av koden til. I større prosjekter vil det være flere iterasjoner i denne fasen. Fasen avsluttes med en første ekstern avlevering av software, milepælen er kalt Initial Operational Capability Milestone.

I transition fasen har produktet forflyttet seg fra utviklingsorganisasjonen til sluttbruker. Aktiviteter i denne fasen er opplæring, testing opp mot sluttbrukers forventninger. Når alle mål er oppnådd er milepælen Product Release Milestone nådd og utviklingsløpet avsluttes.

4.1.4.2.3 Metodens disipliner og arbeidsflyt

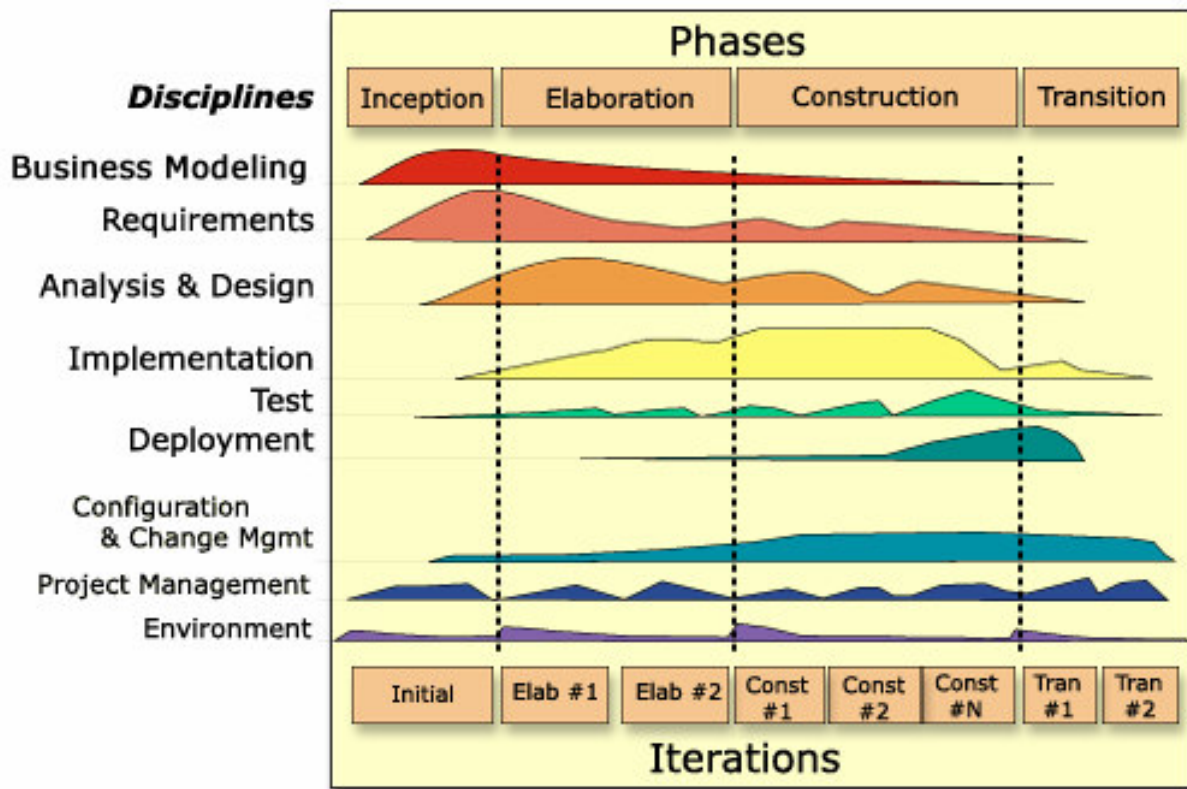
RUP er basert på noen byggestener som beskriver hva som skal lages, nødvendige ferdigheter og en stegvis forklaring av hvordan den forskjellige utviklingsmål i prosjektet skal nås.

Hovedbyggestenene er:

- Roller (who) – En rolle definerer et sett med ferdigheter, kompetanse og ansvar.
- Produkter (Work products) (what) – Produkter er et resultat av en oppgave, herunder dokumentasjon og modeller
- Oppgaver (how) – En oppgave beskriver en arbeidspakke som tildeles en rolle og som gir et meningsfullt resultat

I hver iterasjon blir arbeidsoppgavene fordelt i ni disipliner: Engineering, Virksomhetsmodellering, Kravhåndtering, Analyse og design, Implementering, Test, Overføring samt de støttende disipliner konfigurasjons- og endringshåndtering, prosjektledelse og håndtering av omgivelser.

Figur 11 oppsummerer hovedelementene i RUP, her fremkommer faser, iterasjoner, så vel som disipliner og arbeidsoppgaver og belastning i prosjektløpet.



Figur 11 Rational Unified Process – RUP

4.1.4.2.4 Prosessens anvendbarhet

RUP er en meget veldokumentert prosess, og er også tilrettelagt for spesialisering.

Prosessbeskrivelsen innledes med en forklaring om at nettopp tilpassing er nødvendig⁸. Prosessen er støttet av opplæringspakker, og mange konsulenter har spesialisert seg på å støtte innføring i organisasjoner.

4.1.4.3 Profesjonell Systemutvikling – Død over standardmetodene

Jeg oppsummerer her noen inntrykk fra kurset Systemutvikling – teori og modeller ved Institutt for Informatikk (IFI).

Som student ved systemutviklingsgruppa ved IFI var vi så heldige å få følge hovedkurset Systemutvikling – teori og modeller. Dette kurset forsøker å formidle en tredje vinkling på systemutvikling – en kvalitativ tilnærming.

Dette er unikt har jeg forstått mer og mer i ettertid. Unikt fordi jeg ikke finner så mange metoder på ”markedet” som tar denne tilnærmingen, unik i den forstand at det som blir forelest er at nettopp standardmetoder ikke kan anbefales – men ikke minst unik fordi det er et kurs hvor vi som studenter ble nødt til å tenke selv. Dette skyldes nok ikke bare stoffet alene – med en god blanding mellom teori og svært relevante erfaringer fra virkeligheten, men kanskje mer den muligheten vi fikk for å prøve ut

⁸ RUP – “No single software development process can be applied everywhere!”

våre ideer overfor MEDAKIS-prosjektet ved Rikshospitalet. Den positive erfaringen skyldes nok samlingen av dette med den inspirasjon og det svært gode læringsmiljø som ble skapt av kursansvarlig og foreleser Amanuensis Kristin Braa.

Så etter dette kurset sitter man igjen uten å tro på en spesiell metode – hva har man da? Profesjonell Systemutvikling oppsummerer to hovedpunkter som en oppsummering:

- Aktiv og kompetent ledelse
- Systemutviklere som kan og tør utvikle sine arbeidsformer

Jeg anser det første punktet er noe som kan læres, og hvor god systematisk metodikk vil være til god støtte. Boken angir en god del teknikker, som jeg anser å være godt ivaretatt også i andre systemutviklingsmetoder.

Det første punktet knytter seg til læring – både læring på individnivå og ikke minst læring i organisasjoner. Boken er ganske kort på dette, men har svært gjenkjennbare beskrivelser slik jeg ser det fra en stor utviklingsorganisasjon som min egen. Jeg savner en modell for læring i organisasjoner, og har derfor vært nysgjerrig på dette etterpå.

4.1.5 VERKTØY

Dette skal ikke være noen fullstendig liste over verktøy som finnes på markedet. Jeg gir bare noen enkle eksempler på hva jeg kjenner til, der også de to viktigste verktøyene for oss i Hærens K2IS nevnes.

4.1.5.1 System Engineering verktøy

Requirements Driven Development (RDD) er en softwaresuite som har vært på markedet siden tidlig på 90 tallet. I forbindelse med prosjekt Div 2000 i Hæren tok man i bruk systemutviklingsverktøyet RDD-100. Hærens Forsyningskommando valgte å videreføre bruk av dette verktøyet under oppstart av prosjekt EDBST. Etter hvert oppgraderte man til RDD Com⁹. Verktøyet er omtalt i blant annet hovedoppgaver fra tidlige Norges Tekniske Høyskole. Bruken av verktøyet var godt beskrevet for bruken ved Våpenskolen for Hærens Samband. Verktøyet benyttes i dag ikke lenger i utviklingen av Hærens K2IS, men så vidt jeg kjenner til er det i utstrakt bruk for kravhåndtering andre steder i Forsvaret.

4.1.5.2 UML-verktøy

Systemutviklingsverktøyet Rational Rose¹⁰ er i dag dominerende på markedet. Verktøyet er laget for å understøtte Rational Unified Process.. Verktøyet benytter Unified Modeling Language. Rational Rose er det verktøyet som opprinnelig ble anbefalt til MACCIS, og benyttes i dag av svært mange av Forsvarets leverandører samt i standardiseringsprogrammet MIP. Verktøyet er et kommersielt produkt som også har en relativt betydelig kostnadsside, men har gode opplæringspakker og mange godt trente instruktører. Enterprise Architect¹¹ er et annet UML-verktøy som på grunn av en noe annen kostnadsprofil etter hvert er tatt i bruk for modellering i Forsvaret.

⁹ RDD verktøy, se <http://www.holagent.com/products/product1.html>

¹⁰ Rational Rose, se <http://www.rational.com/>

¹¹ Enterprise Architect, se <http://www.sparxsystems.com.au/>

4.1.5.3 Andre verktøy

Verktøy for Kvalitative metoder har jeg ikke funnet eksempler å vise til, ei heller noen suksesshistorier. Vi som har gjennomført kurset IN364 har opparbeidet noen egne erfaringer fra vår deltakelse i MEDAKIS. Noe spesielt verktøy var det ikke snakk om. Her ble innføring i forskningsmetodikk der kvalitative metoder forklares fokusert.

Et søk på Internett etter kvalitative metoder og verktøy gir relativt mange treff med fokus på den kvalitative metoden som verktøy, men få på software verktøy som støtter metoden. De treffene jeg fikk^{Søk KvalMe} viste til bruk av dataverktøy for å understøtte innsamling og håndtering data i forskningen. Jeg forstår det slikt at dette i dag foreløpig er en metode for å tilnærme seg forskning på et område og at det ikke er noen spesielle kommersielle verktøy som er markedsledende.

4.2 RAMMEVERK

Hva er rammeverk i forhold til utviklingsmodeller, metoder og verktøy? Jeg har ikke hørt noe om rammeverk i forbindelse med systemutviklingsfag ved IFI. I kurset "Modellering med objekter"^[IN-MMO] benyttet vi delvis metoden i Catalysis^[Catalysis]. Metoden foreskriver 3 trinn for å lage ønsket system. Arkitekturbeskrivelse med grensesnitt, komponentbygging og integrasjon. Kurset gikk også ved hjelp av gode eksempler inn på hvordan dette skulle gjøres. Her ble da også begrepet rammeverk introdusert. Dette rammeverket var Referansemodellen for Åpne Distribuerte systemer^[RM-ODP]. Hvis vi studerer dette nærmere ser vi at rammeverket forholder seg forholdsvis uavhengig til utviklingsmodeller, metoder og verktøy. Rammeverket RM-ODP beskriver et system - eksisterende eller som skal lages; og kan dermed oppfattes som en systemmodell. Men hvis man velger å benytte rammeverket for å beskrive et nytt system – sammen med et anbefalt språk og verktøy gir dette føringer for kjeden utviklingsmodell, metode og verktøy.

Det finnes også andre rammeverk. I vårt militære system finnes C4ISR- et rammeverk laget av det amerikanske forsvaret. Dette rammeverket påvirker også NATO. Det later til at NATO benytter C4ISR som basis for å utvikle sin egen NATO C3 Tekniske Arkitektur^[NC3TA]. I det følgende presenteres de to rammeverkene ISO RM-ODP og C4ISR.

4.2.1 ISO RM-ODP

RM-ODP utvikles i felles innsats mellom ISO og ITU-T. Det er den stadig økende utbredelsen av distribuert systemer som har vært motivasjonen bak utviklingen av et rammeverk.

4.2.1.1 Målsetting

Målsettingen med RM-ODP er å oppnå:

- Flyttbare applikasjoner mellom forskjellige plattformer

Samvirke mellom ODP Systemer, med dette menes meningsfull informasjonsutveksling og tilpasset gjenbruk av funksjonalitet over et distribuert system

Distribusjon ved å skjule konsekvensene av distribusjon for applikasjonene, programmerer og bruker.

Referansemodellen gir et oversiktsbilde som organiserer bitene i et ODP system inn i et samspillende hele. Referansemodellen prøver verken å standardisere komponentene i systemet ei heller unødigg å påvirke valget av teknologi^[DSTC]

Rammeverket er ment å standardisere beskrivelsen av åpne distribuerte systemer. Rammeverket for åpne systemer gir en arkitektur hvor det gis støtte for distribusjon, samarbeide, interoperabilitet og portabilitet.

4.2.1.2 Hovedidé

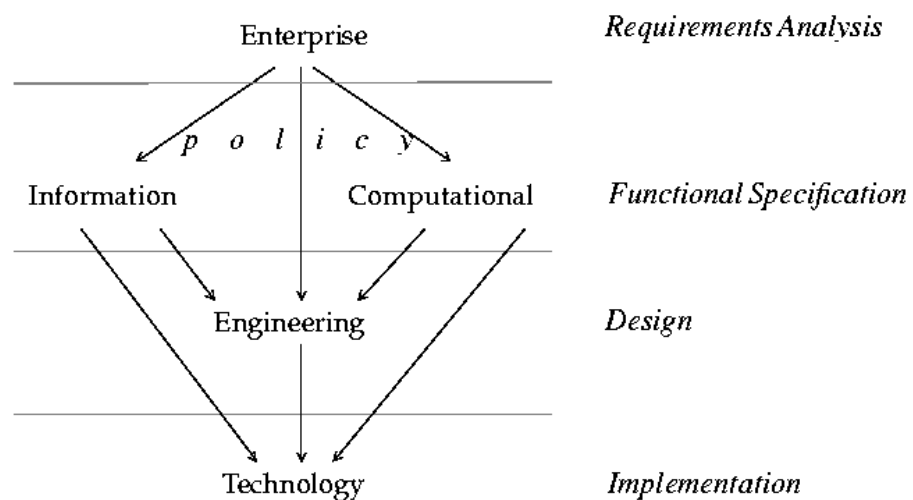
Hovedidéen er å beskrive uavhengige aspekter uavhengig av hverandre, samtidig som det gjøre tydelig hvordan disse uavhengige aspektene legger føringer for hverandre. Strukturen baserer seg på to tilnærminger, dette er "Synsvinkler" og "Gjennomsiktighet".

Med gjennomsiktighet eller transparens menes det at enkelte aspekter ved objekter skjules. Dette gjelder slikt som feilsituasjoner, lokalisering av objekter og replisering.

Med synsvinkler eller viewpoints menes synsvinkler som man ser på systemet fra. Det er definert fem slike viewpoints, det er:

- Enterprise - hensikt, målsettinger og policy
- Information - regler for informasjonslagring og håndtering
- Computational - funksjonell nedbrytning
- Engineering - nødvendig infrastruktur for å understøtte distribusjon
- Technology - teknologiske valg

Anvendelsen av disse fem viewpoints kan i forhold til software engineering illustreres som i Figur 9, RM-ODP synsvinkler og Software Engineering.



Figur 12 RM-ODP synsvinkler og Software Engineering

4.2.1.3 De 5 synsvinklne

4.2.1.3.1 Enterprise viewpoint - virksomheten.

Dette ståsted er organisatoriske krav og struktur. Fra et virksomhetsperspektiv kan sosiale og organisatoriske forhold beskrives med objekter - både aktive så som avdelingssjefen, gruppesjefen, økonomiansvarlig, kunden og passive så som budsjett, regnskap, penger.

- Grupper - grupperinger av objekter som har en felles oppgave, f.eks. kontor, prosjekt e.l.
- Roller til objektene i gruppen, uttrykt som
- tillatelse - sjefen har tillatelse til å godkjenne utbetaling av penger
- forbud - økonomiansvarlig kan ikke utbetale penger det ikke er budsjettet pålegg - den enkelte ansatt skal umiddelbart rapportere feil i lønnsutbetalingen

Virksomhetsbeskrivelsen skal hovedsakelig konsentreres om det som er utførende handlinger som kan endre policy, så som å gi et pålegg eller nekte en tillatelse. Dette innebærer at det å sjekke lønnsutbetaling ikke er nødvendig å beskrive sett fra dette ståstedet. Dette overlates til Computational viewpoint.

4.2.1.3.2 Information viewpoint

Dette ståsted er konsentrert om informasjonen, hvordan denne deles opp og struktureres.

Information viewpoint er en

- modell av informasjonen – med betydning eller semantikk
- sammen med informasjonshåndteringen som endrer informasjonen
- og hvor informasjonen flyter i organisasjonen.

4.2.1.3.3 Computational viewpoint

Dette ståsted beskriver systemet funksjonelt nedbrutt i samvirkende objekter som spiller mot hverandre over definerte grensesnitt.

4.2.1.3.4 Engineering viewpoint

Dette ståsted beskriver hvordan objektene samvirker distribuert. Dette innebærer at den infrastruktur som tillater distribusjon må beskrives her. Det viktigste er å støtte samvirke mellom objektene definert i computational viewpoint. Her beskrives i tilstrekkelig detalj hvordan objektene skal kommunisere via kanaler, hvordan objektene samles i klustere og hvordan disse håndteres i fysiske noder. Dette ståstedet gir en struktur og orden som muliggjør administrasjon av et ODP-system.

4.2.1.3.5 Technical viewpoint

Her beskrives den underliggende infrastruktur i form av hardware og software, med referanse til gjeldende standarder som benyttes. Enkelte egenskaper vektlegges, her er noen eksempler:

- Gjennomsiktighet – transparens
Software spesifikasjoner bør være uavhengig av implementasjonsproblemer som fysisk distribusjon, heterogene plattformer og forskjellige virksomhetsområder. Målet med

gjennomsiktighet er å overlate applikasjonsutviklerens problemer med distribusjon til infrastrukturen. RM-ODP definerer et sett med tjenester for å oppnå gjennomsiktighet, dette er

- Tilgangskontroll (Access transparency) – som skjuler datarepresentasjon og mekanismer for tilgang mellom systemer.
- Feilsituasjoner (Failure transparency) – som skjuler feilsituasjoner og det at objekter gjenoppstår uten av dette skal påvirke systemet
- Lokalisering (Location transparency) – som skjuler det at en applikasjon trenger å vite hvor en tjeneste befinner seg for å kunne kalle på denne.
- Migrering (Migration transparency) – som skjuler det at systemet kan flytte på et objekt. Dette brukes ofte for å bedre utnytte tilgjengelige ressurser og øke tilgjengeligheten på systemet.
- Relokering (Relocation transparency) – som skjuler det at et grensesnitt kan flytte fra andre grensesnitt som er avhengig av det første.
- Replikasjon (Replications transparency) – som skjuler det at det kan eksistere flere kopier av et objekt for å ha stabilitet, tilgjengelighet eller økt ytelse.
- Persistens (Persistence transparency) – som skjuler for et objekt det at andre objekter oppstår eller forsvinner.
- Transaksjon (Transaction transparency) – som skjuler problemer med koordinering mellom aktivitetene i en gruppe av objekter.

4.2.1.4 Anvendelse

RM-ODP er et rammeverk - en referansemodell, ikke en implementasjonsstandard. Denne forskjellen kan være vanskelig å se og forstå. RM-ODP er ofte referert, dette kan man ser hvis man gjennomfører et søk på nettet. Forankring i standardiseringsorganene ISO og ITU er et godt utgangspunkt for at standarden skal utvikle seg.

RM-ODP sier mye om hva som skal beskrives, men sier svært lite om hvordan dette skal gjøres, dvs. rekkefølge og fremgangsmåte for å komme frem til gode resultater. Man kan vanskelig ta for seg RM-ODP og lage et system som etter en kokebok. Dette gjør at mange finner det vanskelig å se hvor man skal starte når RM-ODP skal tas i bruk.

4.2.2 C4ISR-AF

4.2.2.1 Målsetting

Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance Architecture Framework C4ISR-AF^[C4ISR-AF] er et rammeverk utviklet av det amerikanske forsvaret. Hovedmålet med dette omfattende arbeidet var å definere et samlet opplegg “et rammeverk” for kommando, kontroll, kommunikasjon, komputere, etterretning overvåkning og oppklaring (C4ISR)¹² arkitekturutvikling, presentasjon, og integrering. Arbeidet med å utvikle rammeverket har vært en evolusjonær utvikling. Første utgave av et definert rammeverk, versjon 1.0 ble utgitt 07. juni 1996. Den siste utgivelsen, versjon 2.0 som kom ut 18. desember 1997 er en utvidelse og en forbedring av versjon 1.0 basert på tilbakemeldinger fra praktiske bruk.

¹² Dette er noen hovedfunksjoner innenfor den militære virksomheten. Kommunikasjon og komputere kan sies å naturlig høre naturlig innenfor domenet Informasjonssystemer. De andre funksjonene kommando, kontroll, Etterretning, Overvåkning og Oppklaring har tradisjonelt vært kommunikasjonsintensive funksjoner. Med innføring av komputere var de på disse områdene man mente å finne det viktigste potensialet. Derfor er alle disse funksjonene relatert til informasjonsteknologi i det militære domenet.

4.2.2.2 Hensikt

C4ISR–AF (Architecture Framework) har til hensikt å gjøre arkitektur utviklet av geografisk og funksjonelt spredde enheter, kommandoer, grener og enheter tilknyttet det amerikanske forsvaret sammenlignbar. Dette er ment å sikre en ens oppfatning av den enkelte organisasjonens operative virksomhetsarkitektur, deres systemarkitektur og deres tekniske arkitektur samtidig som at dette kan sammenlignes på tvers av forsvarsgrener og i en multinasjonal sammenheng.

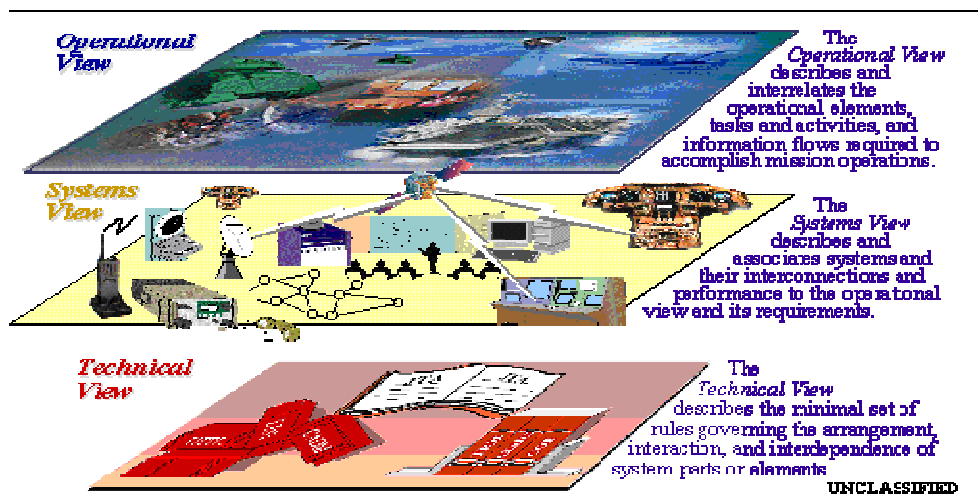
Pålegg fra de bevilgende myndigheter hadde gjennom lovgivning i USA de seneste årene satt fokus på interoperable, integrerte og kosteffektive løsninger for virksomheten innenfor Department of Defence (DoD). Dette er gjort med særlig fokus på informasjonsteknologi. To lover har hatt særlig føring på DoD arkitektur, dette er ITMRA og GPRA¹³ og sammen setter disse lovene krav til at det skal settes felles mål innenfor virksomheten for effektivitet og interoperabilitet i alle enheter av DoD.

ITMRA og GPRA krever at alle enheter innefor DoD måler ytelse på eksisterende og planlagte informasjonssystemer, og rapporterer dette på årlig basis.

C4ISR presenteres da som en felles metode for å beskrive systemene og deres ytelse sett i lys av hvilke oppgaver de kan støtte og hvor godt de gjør dette. Slik kan man si at når rammeverket nå er etablert vil man opprettholde dette for å sikre at man har kontroll over investeringene på IT-siden.

4.2.2.3 De tre synsvinklene

De tre synsvinklene, operativ, system og teknisk beskriver tre sider ved et sammensatt system. Denne sammenhengen kan illustreres slik som i Figur 13 C4ISR - Arkitekturrammeverk - tre synsvinkler på en arkitektur.



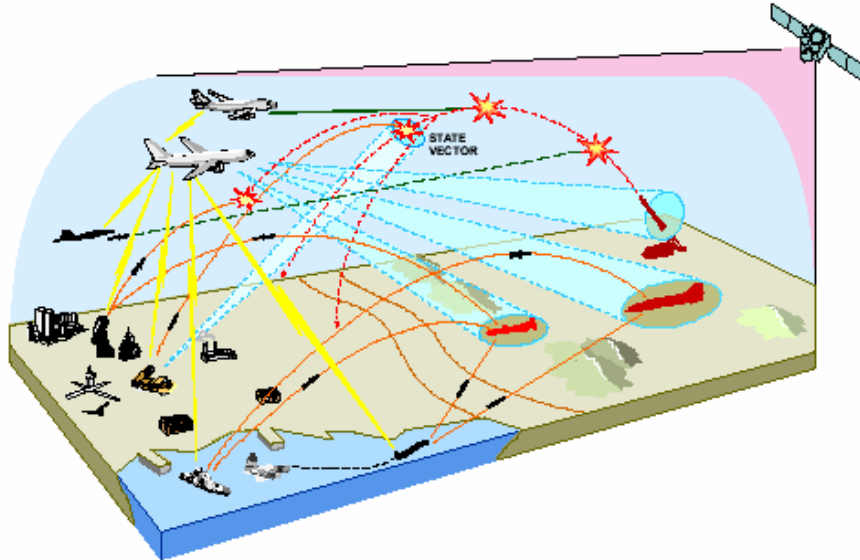
Figur 13 C4ISR - Arkitekturrammeverk - tre synsvinkler på en arkitektur

4.2.2.3.1 Operativ synsvinkel

Den operative synsvinkelen på arkitektur er en beskrivelse av oppgaver og aktiviteter, operative enheter og informasjonsflyt som er nødvendig for å gjennomføre og støtte en militær operasjon.

¹³ ITMRA - Information Technology Management Reform Act. Denne loven er også kjent som Clinger-Cohen act, denne loven kan man finne mange referanser til innefor DoD. GPRA - Government Performance and Result Act

Den består av beskrivelser som gjerne kan være grafiske fremstillinger. Dette er operative elementer, tildelte oppgaver og aktiviteter og informasjonsflyt som støtter de som faktisk utfører oppgaven (The Warfigther) Et eksempel på en overordnet beskrivelse, er vist i Figur 14.

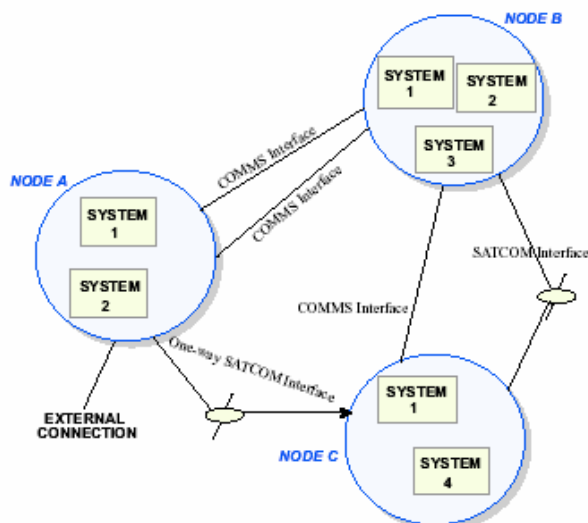


Figur 14 Overordnet bilde operativ kontekst (OV-1) - Eksempel fra luftvern

4.2.2.3.2 Synsvinkel system

Synsvinkel system på arkitektur er en beskrivelse av systemer og forbindelser som gir eller støtter operative funksjoner (war fighting functions) Dette kan på et overordnet nivå innefor en virksomhet vise hvordan flere systemer er koblet og virker sammen. På et lavere nivå kan den for et enkelt system beskrive intern oppbygging og tjenester til den enkelte enhet i arkitekturen. For et system vil denne synsvinkelen vise fysiske forbindelser, plassering og identifisering av de viktigste noder, kretser, linjer, nettverk og plattformer dette er plassert på. Dette bør også identifisere essensielle ytelsesparametre som MTBF^{MTBF}, vedlikeholdbarhet og tilgjengelighet. Synsvinkel system skal vise sammenhengen mellom fysiske enheter og deres ytelse til den operative arkitektur og de krav som der er stilt. Et eksempel på en beskrivelse som viser grensesnitt mellom deler i et system er gitt i Figur 15 Systemgrensesnittbeskrivelse SV-1.

MTBF Mean Time Between Failure



Figur 15 Systemgrensesnittbeskrivelse SV-1

4.2.2.3.3 Teknisk synsvinkel

Den tekniske synsvinkel på arkitektur er et minimalt sett av regler for å styre plassering, samhandling og interne avhengigheter mellom systemer og delelementer. Dette skal sikre at de helhetlige system virkelig tilfredsstiller de gitte operative krav.

Den tekniske synsvinkel gir retningslinjer som senere brukes for tekniske implementering av systemet, for å lage spesifikasjoner og for å identifisere enheter i systemet. I den tekniske synsvinkel inngår et utvalg av tekniske standarder som skal anvendes sammen med regler og kriterier for å gruppere tjenester og grensesnitt. Dette kalles for en teknisk profil. Et eksempel på en slik profil er gitt i Figur 16 Teknisk arkitekturprofil (TV-1).

Service Area	Service	Standard
Operating System	Kernel	FIPS Pub 151-1 (POSIX.1)
	Shell and Utilities	IEEE P1003.2
Software Engineering Services	Programming Languages	FIPS Pub 119 (ADA)
User Interface	Client Server Operations	FIPS Pub 158 (X-Window System)
	Object Definition and Management	DoD Human Computer Interface Style Guide
	Window Management	FIPS Pub 158 (X-Window System)
	Dialogue Support	Project Standard
Data Management	Data Management	FIPS Pub 127-2 (SQL)
Data Interchange	Data Interchange	FIPS Pub 152 (SGML)
	Electronic Data Interchange	FIPS Pub 161 (EDI)
Graphics	Graphics	FIPS Pub 153 (PHIGS)
. . .		

Figur 16 Teknisk arkitekturprofil (TV-1)

4.2.2.4 Retningslinjer, prosess og støtteverktøy

C4ISR-AF gir fire typer støtte for arkitekturutviklingsprosessen.

Dette er

- 1) Retningslinjer med hovedprinsippene for å bygge arkitekturer som i overensstemmelse med rammeverket
- 2) En prosess for å bruke rammeverket til å bygge og integrere arkitekturer
- 3) En gjennomgang av arkitekturdata og verktøy som kan lette arkitekturbeskrivelsesprosessen
- 4) En detaljert beskrivelse av produktene

De tre første typene gjennomgås i dette kapittelet, produktene omtales nedenfor i kapittel 4.2.2.5.

4.2.2.4.1 Hovedprinsipper

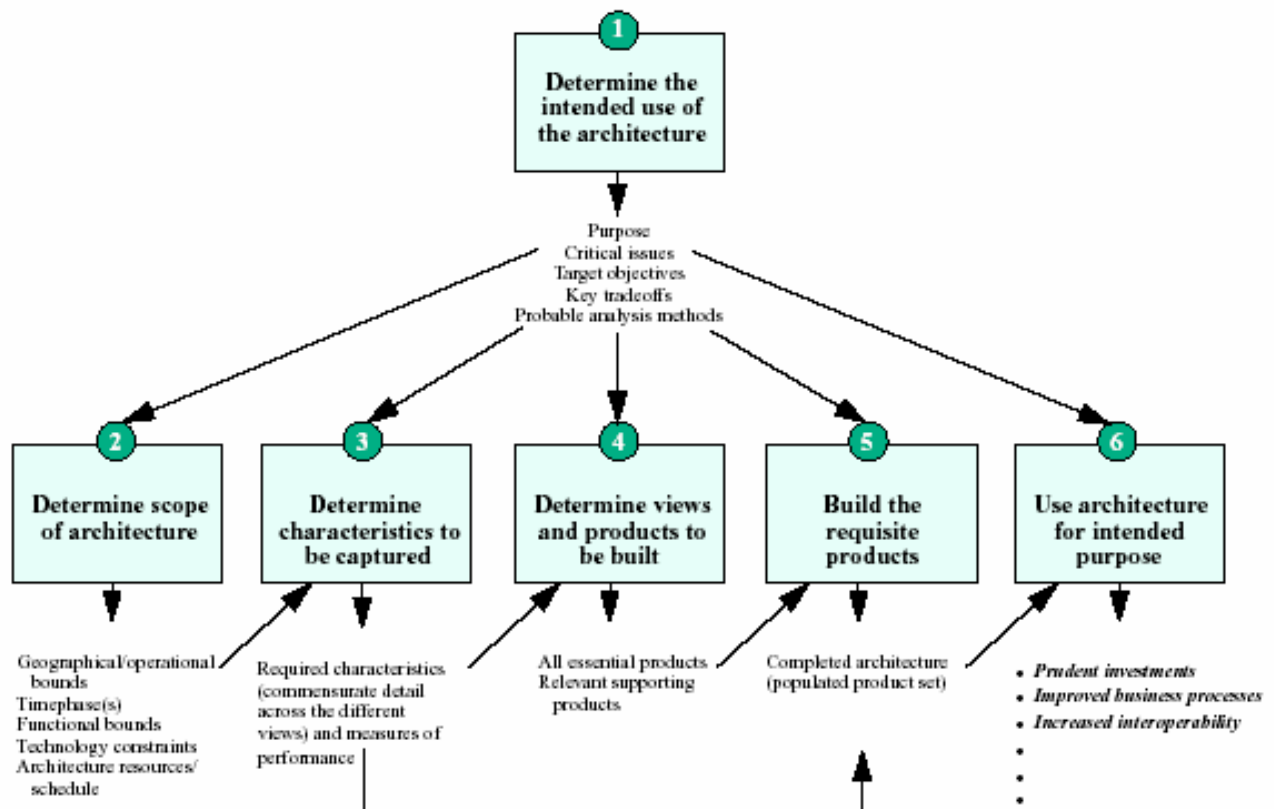
C4ISR-AF fremfører som hovedprinsipper at arkitekturer skal bygges med en helt klar hensikt. Dette bør være en ledetråd under hele prosessen

Arkitekturer skal videre lette, ikke hindre kommunikasjons mellom mennesker. Det anbefales her er å bruke grafiske fremstillinger

Siste men ikke minst skal arkitekturer skal kunne relateres, sammenlignes og integreres i hele DoD

4.2.2.4.2 Arkitekturbeskrivelsesprosess

De grunnleggende steg for å bygge en arkitektur i henhold til rammeverket er meget kort beskrevet. Det er vist hvordan man i de fleste tilfeller vil gå frem for å komme til de vesentligste produktene. Det viktigste hjelpemidlet er en illustrasjon som gir en rettesnor for prosessen, se Figur 17 En sekstrinns prosess for å bygge arkitekturer.



Figur 17 En sekstrinns prosess for å bygge arkitekturer

4.2.2.4.3 Hjelpemidler – arkitekturdata og verktøy

Før rammeverket kom til brukte den enkelte enhet innenfor DoD sine egne metoder for å lage C4ISR arkitekturer. Arkitekturdatabaser som ble benyttet var laget rundt sin egen datamodell.

Dette gjorde det vanskelig å utveksle informasjon mellom systemutviklere, og dermed sikre interoperabilitet.

Med C4ISR-AF på plass er det nå ønskelig med en felles måte for å organisere og beskrive informasjonen i arkiturene strukturert.

4.2.2.4.3.1 CADM

C4ISR Core Architecture Data Model (CADM), er en formell modell med arkitekturprodukter, deres struktur og deres relasjoner. CADM har til hensikt å være en felles metamodel eller logisk skjema som kan ta vare på holde informasjonen i arkiturenes.

CADM tilbys som et verktøy for å lette felles analyse, og bistå med å sjekke kompletthet og konsistens i og mellom arkitekturer.

CADM er ikke i seg selv en del av C4ISR-AF og CADM vil først og fremst være nyttig for de som lager verktøy til støtte for å lage arkitekturer.

4.2.2.4.3.2 *LISI referansemodell*

Det å beskrive hvordan systemer settes sammen for å oppnå et felles mål er en de største utfordringene i dag for de som skal utvikle arkitekturer for informasjonssystemer innefor DoD. Det å forstå hva det er og grad av interoperabilitet er et nøkkelement. Dette må være med når man beskriver så vel som designer, utvikler eller tar i bruk en arkitektur for et informasjonssystem. "The Levels of Information Systems Interoperability (LISI) Reference Model" gir en logisk struktur. Den kan også brukes som en modell for modenhet for å øke nivået av interoperabilitet mellom informasjonssystemer gradvis. LISI er nærmere beskrevet i kapittel 4.3.

4.2.2.5 Produkter

Innefor hver synsvinkel beskriver C4ISR-AF et sett av produkter. Disse nummereres fra 1 og oppover, for eksempel OV-1, SV-1 og TV-1, innenfor henholdsvis Operativ synsvinkel, System synsvinkel og Teknisk synsvinkel. Mange systemer er allerede i bruk i dag, og er dermed beskrevet på sin måte. For å kunne sette nye systemer inn i en kontekst av disse gamle systemene, er det gitt et minimum av beskrivelser som skal følge alle systemer, både gamle og nye.

4.2.2.5.1 Obligatoriske produkter og tilleggsprodukter

C4ISR-AF presenterer felles prinsipper og teknikker som kan brukes av organisasjoner på alle nivåer for å bygge arkitekturer. Det vesentlige forblir likevel å kunne lage arkitekturer som kan brukes i fellesoperative og multinasjonale analyser. To viktige scenarier her er (1) den raske "klar til strid" og (2) analysen som underbygger langsiktige investeringer for DoD. Begge disse analysene må gi arkitekturer som er sammenlignbare og som kan settes sammen. For å sikre at enhver arkitektur kan brukes i en slike felles analyse må alle arkitekturer ha et felles sett av standard produkter. Derfor deler man inn produktene i to kategorier, nemlig

- Essensielle produkter

Disse produktene er det minimum sett av beskrivelser for å kunne utvikle arkitekturer som kan forstås på en felles måte og som kan brukes på tvers innefor hele DoD og mot multinasjonale elementer

- Tilleggsprodukter

Dette er produkter som gir informasjon som kan være nødvendig avhengig av hensikt og mål for en mer spesifikk arkitektur. Det er den hensikt man ønsker å oppnå med arkitekturen som bestemmer hvilke av disse tilleggsproduktene man ønsker å utvikle.

4.2.2.6 Anvendelse

Rammeverket forteller hvordan man skal beskrive arkitekturer, men rammeverket beskriver ikke hvordan man skal utvikle eller implementere en spesiell arkitektur. Rammeverket beskriver heller ikke hvordan man skal utvikle eller anskaffe systemer eller systemer som skal virke i en sammenheng (system of systems).

Dette skillet mellom arkitekturbeskrivelse og arkitekturimplementering er grunnleggende. Arkitekturbeskrivelsen er som en gjennomarbeidet plan (blueprint) for en eksisterende eller planlagt sammensetting av ressurser, regler og relasjoner. Når denne planen tas med over i konstruksjons- og implementerings- og anskaffelsesfasen av et system blir beskrivelsene også overført til kapasiteter og beskrivelser som passer for disse formålene. Rammeverket C4ISR-AF tar ikke for seg denne overgangen fra plan til implementering.

4.2.2.7 Oppsummering

Etter at C4ISR-AF ble gjort tilgjengelig i desember 1997 ble det utarbeidet en termineringsrapport [C4ISR-FR]

Hovedkonklusjonen i denne rapporten er

”The Need – A Unified C4ISR Development Process“

For å kunne få ut i felten systemer som er interoperable, integrerte og som er kosteffektive C4ISR kapasiteter, må DoD etablere en felles prosess. Denne prosessen må ivareta arkitekturaspekter på tvers av forsvargrener. Den må ivareta rammeverk og modeller sammen med retningslinjer for integrasjon, interoperabilitet, måling, testing, prosesser, evalueringsmetoder, ytelsesmålinger. Ideelt sett bør denne prosessen virke slik:

Fortsatt utvikling av C4ISR arkitekturer fortsetter på en distribuert måte som i dag og Arkitekturene kan på en lett måte sammenlignes på tvers av organisatoriske skillelinjer.

DoD vil kunne sammenligne disse integrerbare arkitekturene for å:

Avklare og avstemme forskjeller for å sikre et godt felleoperativt samspill

Vurdere nytteverdi i systemer basert på dagens og fremtidig teknologi

Nyttiggjøre seg nye muligheter

Identifisere og prioritere hensiktsmessig interoperabilitet mellom nøkkelsystemer

Vanskelige konsepter kan testes på et tidlig stadium for å se om de kan holde og om de vil være kosteffektive før går til en kostbar anskaffelse eller storskala utprøving

Ideer, nye muligheter, mindre prøver og utprøvde løsninger i felt kan føres tilbake gjennom revisjon av arkitekturen for å vurdere mulig påvirkning på effektiviteten på fremtidige operasjoner

Denne ideelle prosessen er forsøkt illustrert på denne måten, se Figur 18



Figur 18 En felles prosess for å oppnå en interoperabel og kosteffektiv C4ISR dugelighet

4.2.2.7.1 Hva er hindringene?

I dag¹⁴ er det flere hindringer som må overkommes for at DoD skal kunne ta i bruk en felles C4ISR prosess. Først sies det å være manglende retningslinjer for arkitekturutvikling innenfor organisasjonen slik at produkter kan sammenlignes på tvers. Videre er definisjonen av interoperabilitet for avgrenset. Det listes opp "grader" av interoperabilitet, men det sies for lite om hva den enkelte grad består i. For å få til en enkel avklaring på dette og for dermed å kunne oppnå større grad av interoperabilitet, må det beskrives på en oversiktlig måte. Sist sier termineringsrapporten at det er få eller ingen felles praktiske måter og prosedyrer for å integrere, eksperimentere og teste interoperabilitet. Dette siste kan oppfatte som en kritikk av LISI. Jeg mener selv i denne oppgaven at LISI beskriver interoperabilitet på en hensiktsmessig måte. Det kan være at det LISI oppfattes som noe teoretisk, og at det er de manglende praktiske sidene som mangler. Dette skyldes vel blant annet at dette er ganske nytt for mange av de som er ment å gjøre jobben med å integrere og test. Da er nok ikke dette en kritikk direkte av LISI, det er heller slik at man trenger mer tid for å få dette tatt i bruk og modnet slik at det kan bli en del av organisasjonen.

¹⁴ Dette er basert på sluttrapporten [C4ISR-FR] datert i 1998.



Figur 19 De manglende elementene for å oppnå en felles prosess

Dette gjør at DoD i dag fortsatt ikke har én måte for å se sammenhengen mellom bruk av økonomiske virkemidler og operativ C4ISR dugelighet. De manglende elementene for å oppnå en felles prosess er illustrert i Figur 19.

4.2.2.7.2 Bidrag fra AWG - Hva er oppnådd?

Til tross for de hindringer som er skissert, har C4ISR-AF, likevel brakt DoD videre. C4ISR Architecture Working Group, AWG, er de som har gjort denne jobben. De har levert produktene sammen med anbefalinger som fremkommer i rapporten, og dette vil bidra til å lette det videre arbeidet. Målet er nå å komme over de hindringene som ligger der for å oppnå C4ISR interoperabilitet og kosteffektivitet. Ved å videreføre utviklingen av produktene på en evolusjonær måte vil produktene utvikle seg.. Produktene er anvendbare i dag, og det virker som om organisasjonene innefor US DoD jobber etter intensjonene i rammeverket. Et søk etter C4ISR på Altavista gir over 10.000 websider, dette illustrer til en viss grad at rammeverket fortsatt er svært aktuelt.

4.3 INTEROPERABILITET

Interoperabilitet - samspilleevne¹⁵; ordet bør ha en god klang i våre ører. Definert som samspilleevne gir interoperabilitet flere assosiasjoner, som i et orkester skal flere spille sammen - man spiller riktig, - man spiller etter en plan, - man treffer tonene samtidig - og sammen skal det ha en helhetlig klang. Vi erkjenner når det gjelder en musikkomposisjon at en komponist har komponert, at en dirigent dirigerer og at en førstefiolinist setter tonen - hvordan er dette innefor det informatikkfaglige?

Denne hovedoppgaven er skrevet ut fra en kontekst hvor det er de militære behov som skal etterstrebes. Når vi nå ser på begrepet interoperabilitet, tas det utgangspunkt i en NATO ramme.

4.3.1 BEGREPSAVKLARING

NATO har definert "interoperabilitet" i [AAP-6] *"Systemers, enheters eller styrkers evne til å tilby tjenester og til å akseptere tjenester fra andres systemer, enheter eller styrker, og å bruke disse tjenestene på en slik måte at de kan operere effektivt sammen"*¹⁶

Denne definisjonen har etter hvert etablert seg som den gjeldende.

Dessverre har ikke dette vært den eneste definisjonen i NATO. Som flere har påpekt, for eksempel i Per Trygve Gundersen i sin hovedoppgave fra 1998^[PTG,98] sies det at NATO Interoperability Planning Document^[NIPD] benytter den samme definisjonen, men når man skal definere ønsket nivå av interoperabilitet settes dette i en kontekst av Behov for brukermedvirkning, Tilgangs- og sikkerhetsaspekter, og Operative, prosedyremessig, tekniske og/eller andre begrensninger for standarder og deres anvendelse.

I [PTG,98] anses dette som en utvidelse av definisjonen. Han velger derfor å avgrense videre behandling av interoperabilitet i forhold til disse tre aspektene.

Jeg tror det er nødvendig med noen kontekstbegreper, og vil vise det gjennom presentasjon av noen modeller. Grad av interoperabilitet kan benyttes for å velge hvilket samvirke man ønsker innenfor hver kontekst. Jeg går nå nærmere inn i begrepet interoperabilitet.

4.3.1.1 Interoperabilitet - integrert

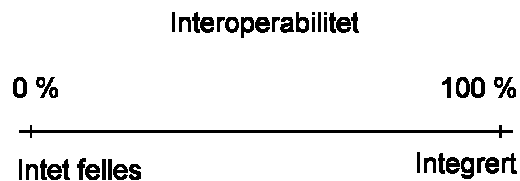
Termen "interoperabilitet" forveksles ofte med "integrert".

Termen integrert refererer også til systemer, prosesser og mekanismer som samhandler. Termene interoperabilitet og integrert er derfor svært like i betydning. Gundersen gjør en god avklaring og jeg låner hans avgrensninger. Det gjøres opp to forståelser av sammenhengen mellom interoperabilitet og integrert. Den første og enkle illustreres¹⁷ best i en figur, se Figur 17, Interoperabilitet - Integrert.

¹⁵ CLUE for Windows versjon 4.2

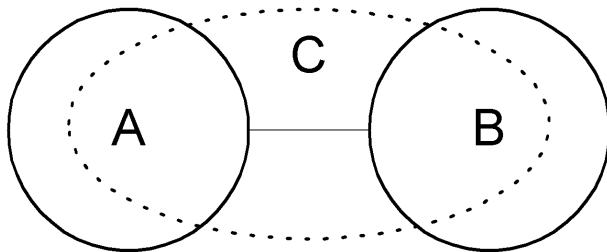
¹⁶ Ordrett: "The ability of systems, units or forces to provide services to and accept services from other systems, units or forces and to use the services so exchanged to enable them to operate effectively together"

¹⁷ Figur 2. Integration as a grade of interoperability



Figur 20, Interoperabilitet - Integrt

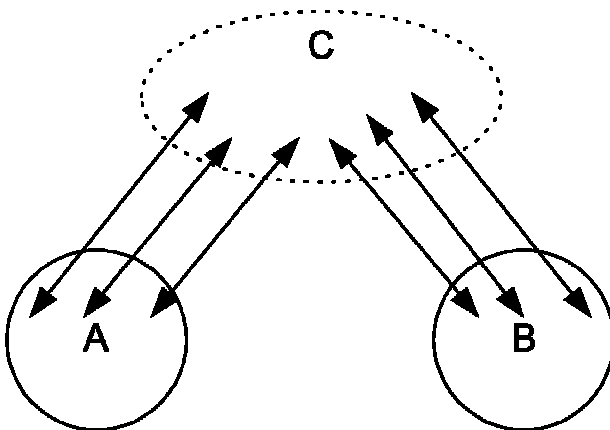
Figur 17 illustrerer at integrt er en inkluderende/ekskluderende egenskap hvor for eksempel "delvis integrt" ikke lenger er en gyldig term. Den andre forståelsen tar utgangspunkt i virksomhetsområder¹⁸. Integrasjon gjelder innenfor ett virksomhetsområde. Interoperabilitet har bare en betydning når man opererer på tvers av virksomhetsområder. Gundersen illustrerer først med en figur hvordan to systemer kan oppfattes som integrt, se Figur 18 Integrasjon av to systemer til ett.



Figur 21 Integrasjon av to systemer til ett

Her er system A og B integrt, det vil si de er like, og selv om de ikke er like kan konseptet C ivareta begrepet integrt sett fra omverden.

Hvis system A og B er forskjellige, bygget for forskjellige formål, altså ikke integrt kommer en annen beskrivelse til anvending. Figur 19 Et felles grensesnitt er hentet fra Gundersens oppgave for å illustrere begrepet interoperabilitet.



Figur 22 Et felles grensesnitt

System A og B er ikke integrt. A og B er interoperable mot grensesnitt C når de har innebygd tilstrekkelig funksjonalitet til å ivareta de felles tjenester som C krever. Gundersen sier så at uten

¹⁸ [PTG98] Hans term "jurisdiction"

standardisering av grensesnitt vil ikke interoperabilitet være mulig etter denne andre forståelsen. Jeg er enig at uten en form for standardisering er ikke interoperabilitet mulig.

4.3.1.2 Interoperabilitet i forskjellige fora

Interoperabilitet som begrep har flere definisjoner, dette skyldes først og fremst at NATO har flere organer som har meninger om dette emnet. Jeg har derfor gjort et utvalg basert på en rapport fra Forsvarets Forskningsinstitutt ^[OPER]. Rapporten kartlegger operative informasjonssystemer i dag samt pågående prosjekter. Interoperabilitet blir i rapporten fokusert mot utveksling av data/informasjon, og man velger å ikke gape over hele problemstillingen rundt interoperabilitet. I rapporten fremheves det at man ikke bør henge seg altfor mye opp i definisjonen av interoperabilitetsnivå, men heller tenke over hva som er hensikten med interoperabilitet. Jeg beskriver i dette kapitlet fire forskjellige modeller som forsøker å klargjøre begrepet og definer nivå av interoperabilitet. Dette er:

- NATOs interoperabilitetsmodell (APP-10)
- NC3-board sin modell
- LISI-modellen
- RM-ODP

4.3.2 TYPER INTEROPERABILITET I NATO

4.3.2.1 Organisatoriske skiller

I NATOs omtale av interoperabilitet har jeg funnet at man opererer med fire typer interoperabilitet

- Vertikal Interoperabilitet
Evnen til å dele informasjon gjennom myndighetsnivå (f.eks. fire nivåer ned i en organisasjon)
- Horisontal Interoperabilitet
Evnene til dele å informasjon mellom forsvarsgrener (f.eks. Hær, Sjø og Luft) mellom funksjonelle områder (f.eks. etterretning og logistikk) og mellom nasjonale styrkeelementer
- Intern Interoperabilitet
Evnen NATO og tilknyttede nasjonale systemer har til å utveksle informasjon både vertikalt og horisontalt.
- Ekstern Interoperabilitet
Evnen NATO og tilknyttede nasjonale systemer har til å gjennomføre utveksling og deling av informasjon med PfP-nasjoner¹⁹ og andre nasjoner som deltar i NATO operasjoner.

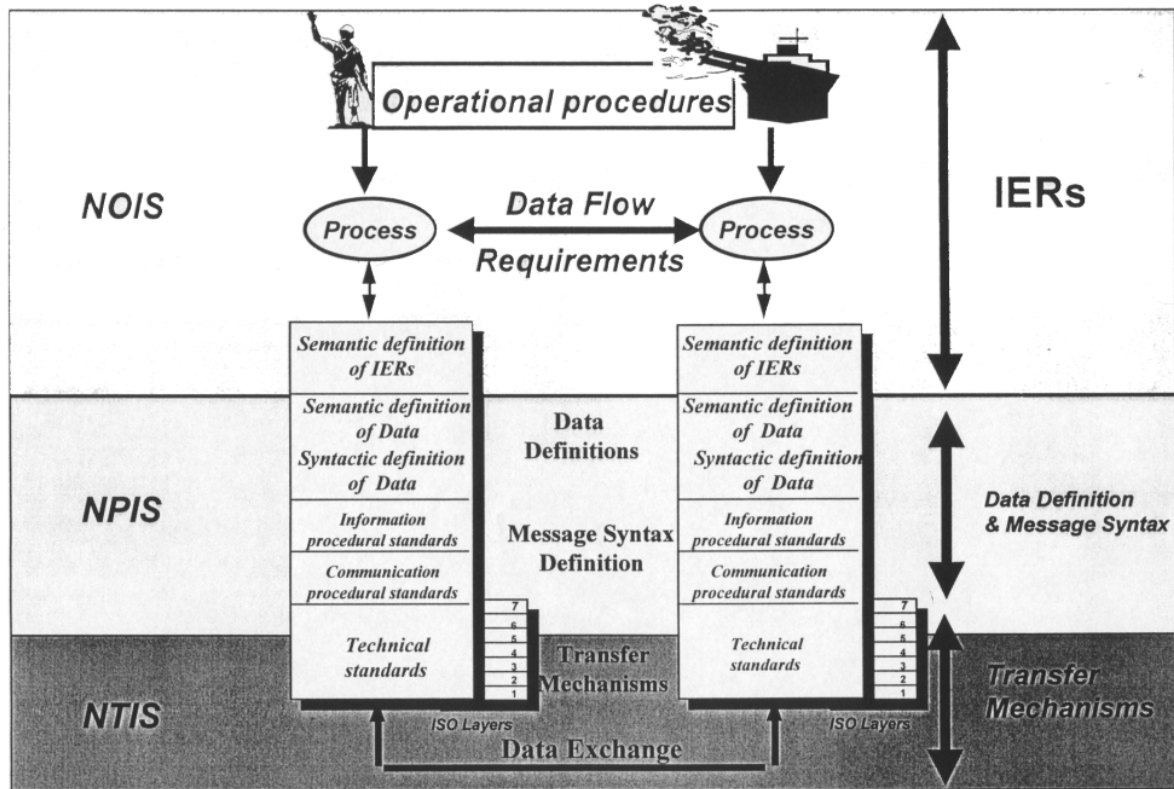
4.3.2.2 Interoperabilitetsstandarder

NATO har gjennom NCIS^[NCIS] beskrevet et sett av standarder som omtaler interoperabilitet. Dette er operative^[NOIS], prosedyremessige^[NPIS] og tekniske^[NTIS] forhold som berører interoperabilitet. Denne

¹⁹ Partners for Peace. En betegnelse som brukes i NATO om ikke-NATO nasjoner som har medvirket under internasjonale operasjoner, for eksempel i tidligere Jugoslavia.

^[NCIS] NATO Common Interoperability Standards

delingen er gjort for en god del år tilbake, men kan lett relateres til et tredelt perspektiv som for eksempel brukes av NATO i dag innenfor NC3TA. En sammenheng mellom de tre forhold er illustrert i Figur 23 Interoperabilitetsstandarder i NATO.



Figur 23 Interoperabilitetsstandarder i NATO

4.3.2.3 Grader av interoperabilitet

I NATO har man basert begrepet på definisjonen gitt i [AAP-6]. For å definere grader av interoperabilitet har man benyttet et noe annet begrep nemlig "levels of interconnection". Som vist nedenfor i kapittel 4.3.3.1 NIPD opererer man i NATO med 6 slike nivåer.

[NOIS] NATO Operational Interoperability Standards

[NPIS] NATO Procedural Interoperability Standards

[NTIS] NATO Technical Interoperability Standards

4.3.3 MODELLER FOR INTEROPERABILITET

4.3.3.1 NIPD

Det defineres seks nivå i NATOs gradering under [NIPD]. Dette er:

NATO Nivå 1:	Enheter og/eller individer utveksler muntlig eller skrevet informasjon vha offline kommunikasjonssystemer
NATO Nivå 2:	Utteksling av muntlig eller skrevet informasjon vha samlokaliserte liaison team, hvor hvert team har en egen terminal til sine egne system.
NATO Nivå 3:	En enkelt operatør overfører informasjon fra et system til et ved å bruke en terminal for hvert system.
NATO Nivå 4:	System til system forbindelse med forhåndsbestemte begrensninger og dynamisk styring av tilgang til data.
NATO Nivå 5:	System til system forbindelse med dynamisk styring av tilgang til data
NATO Nivå 6:	System til system forbindelse med full tilgang til all informasjon og tjenester på begge system.

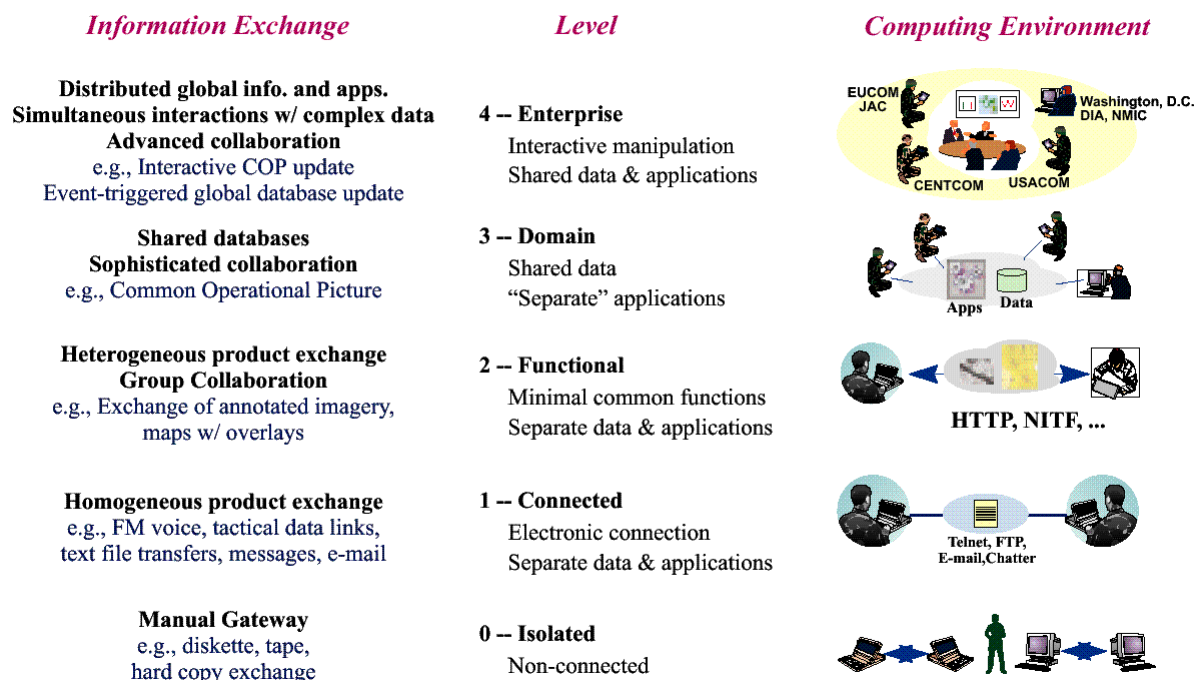
Tabell 3 NIPD 6 interoperabilitetsnivå

Videre slås det fast at når man skal sette krav til grad av interoperabilitet skal man ikke spesifisere operative, prosedyremessige eller tekniske parametre. Dette er implementasjonsmessige detaljer som skal velges i tråd med gjeldende standarder. Det som skal beskrives og argumenteres for fra kravstilleren er ønsket grad av interoperabilitet med hensyn på

- Grad av personell involvert
- Tilgangs og sikkerhetsaspekter, og
- Operative, prosedyremessige og tekniske begrensninger for bruk av standarder eller utvikling av disse.

4.3.3.2 LISI

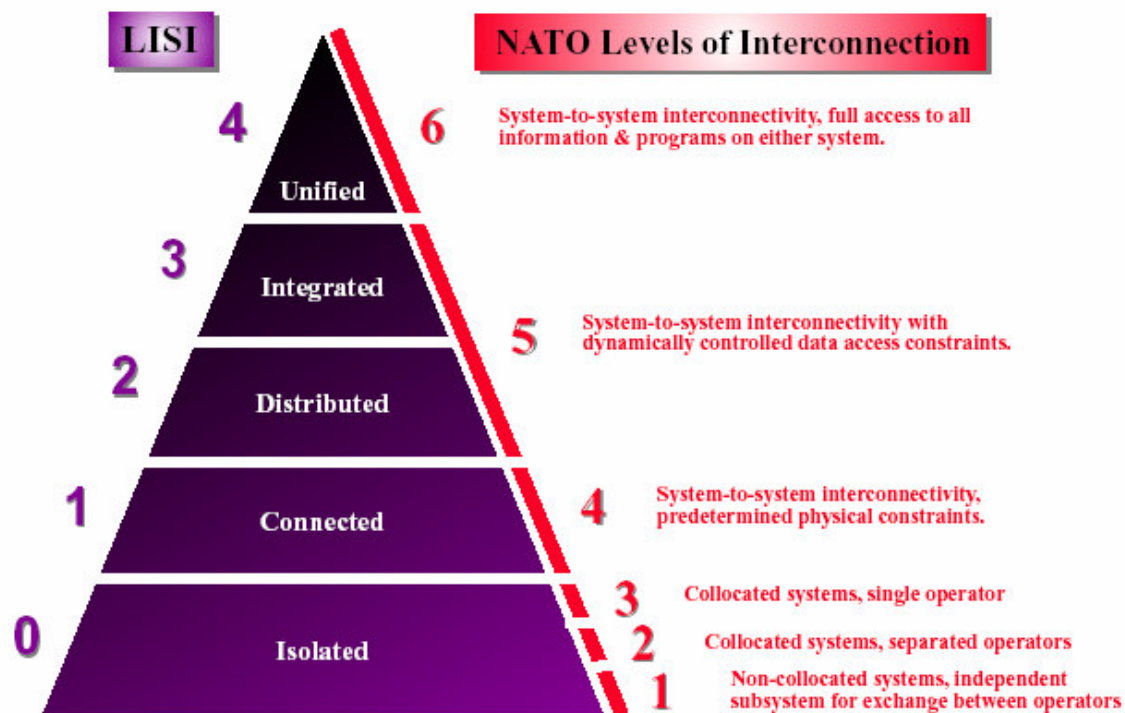
Som en del av arbeidet med å lage C4ISR-AF utført for DoD ble også Levels of Information Systems Interoperability ^[LISI] definert. LISI definerer en modell de kaller modenhetsmodell for interoperabilitet. En illustrasjon som viser teknologi knyttet til de fem definerte nivå av interoperabilitet er vist i Figur 24 LISI Nivå og tilhørende EDB - utrustning. Denne illustrasjonen viser godt inndelingen i fem nivå.



Figur 24 LISI Nivå og tilhørende EDB - utrustning

LISI relaterer seg til NATOs modeller, for som vist i ovenfor i kapittel 4.3.3.1 NIPD har NATO definert inn en person på de tre første nivåene som en del av forbindelsen for å få til interoperabilitet. Disse tre nivåene er pakket sammen i LISI nivå 0 – Isolert. Her er det prosedyrer som er den dominerende faktor slik det også fremkommer i LISI *Capabilities Model*.

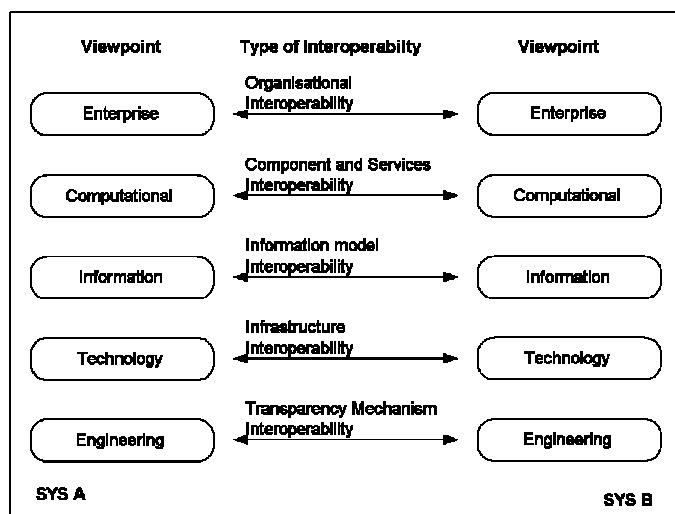
Sammenhengen mellom NATOs grader av interoperabilitet og LISI er vist i Figur 25 Sammenheng mellom NATOs nivå og LISIs nivå.



Figur 25 Sammenheng mellom NATOs nivå og LISIs nivå

4.3.3.3 RM-ODP

ODPs referansemodell som er omtalt i kapittel om systemutvikling, har også en modell for interoperabilitet. Gjennom sine fem perspektiver illustrerer denne modellen at interoperabilitet kan ses på fra flere synsvinkler. For ODPs referansemodell se Figur 26 ODPs referansemodell.

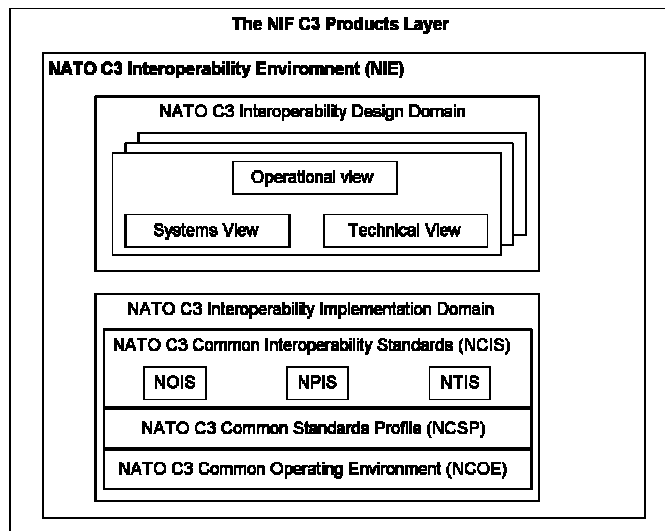


Figur 26 ODPs referansemodell

Denne modeller er også basis for MACCIS som omtales senere i denne oppgaven.

4.3.3.4 NC3TA/NIMP

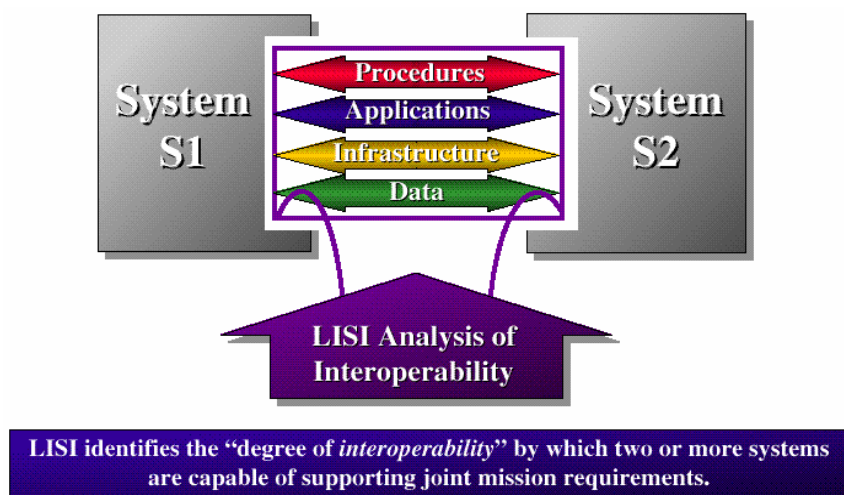
Figur 27 NATO C3 Interoperabilitetsmiljø illustrerte inntil nylig NATOs syn på interoperabilitet.



Figur 27 NATO C3 Interoperabilitetsmiljø

Siden har NC3TA ^[NC3TA] kommet ut i revidert form. Den tok inn i seg NATO Interoperability Management Plan (NIMP)

NIMP (1998) definerer en annen modell som har en litt annen vinkling. NIPD sin skala er sterkt vinklet mot den formen for fysisk forbindelse som det er mellom systemene. NIMP sin modell er vinklet mer mot dataene som faktisk overføres, og hvordan disse kan brukes i systemene. Figur 28 NIMP modell for interoperabilitet viser de forskjellige aspektene som betraktes.



Figur 28 NIMP modell for interoperabilitet

Denne modellen refereres til som NATO C3 Levels of Information Systems Interoperability (NC3 LISI). De forskjellige nivåene i modellen er:

- Nivå 0: Ingen direkte utveksling. Dette innebærer at det ikke er noen fysisk forbindelse mellom systemene, utveksling kan bare skje manuelt.
- Nivå 1: Ustrukturert datautveksling. Dette innebærer utveksling av menneskelesbar ustrukturerte data som f.eks. frieteksten man finner i operasjonsanalyser.
- Nivå 2: Strukturert datautveksling. Dette innebærer utveksling av menneskelesbar strukturerte data som er tilrettelagt for manuell og/eller automatisk behandling, men som krever manuell sammenstilling, mottak og/eller sending av meldinger.
- Nivå 3: Sømløs deling av data. Dette innebærer en automatisk deling av data mellom systemer som er basert på en felles utvekslingsmodell.
- Nivå 4: Sømløs deling av informasjon. Dette er en utvidelse av nivå 3 for å oppnå en universell forståelse av informasjon basert på samvirkende applikasjoner.

Denne modellen er basert på US DoD sin Levels of Information Systems Interoperability (LISI) Reference Model.

4.3.3.5 Oppsummering

Som vist i dette kapitlet finnes det flere modeller. Det grunnleggende er å ta i bruk en modell. For vårt formål kan det se ut som om LISI modellen er det mest hensiktsmessig. Denne er sagt å ligge til grunn for NC3TA. Ved å velge en ambisjon, for eksempel nivå 3 fordrer dette en arkitektonisk ambisjon. Systemene må da være basert på en felles utvekslingsmodell.

4.3.4 INITIATIV FOR Å OPPNÅ INTEROPERABILITET

Det foregår i dag mange initiativ for å forbedre interoperabilitet i NATO. Eksempler er prosedyremessige forhold, utvekslingskrav og tekniske løsninger.

Felles doktriner, felles skoler og utdanning er eksempler på prosedyremessige forhold.

Information Exchange Requirements, i NATO definert i APP-9 er et eksempel på utvekslingskrav. Eksempler på tekniske løsninger er meldingsstandarder og MIP.

MIP er en teknisk løsning basert på felles informasjonsmodell. For ytterligere informasjon se MIPs hjemmeside^[MIP]

4.4 INFORMASJONSINFRASTRUKTUR

Infrastruktur - begrepet forbindes ikke alltid med telekommunikasjon, det er vel så vanlig innen bygg- og anlegg der det beskriver "basistjenester" som veier, vann, kloakk og strøm. Der det er investert enorme summer over utallige år. Og der endringer derfor ikke kan gjøres raskt. Dette er en grunnleggende tanke fra digital kommunikasjons spede begynnelse at dens utbredelse var avhengig av felles internasjonale standarder

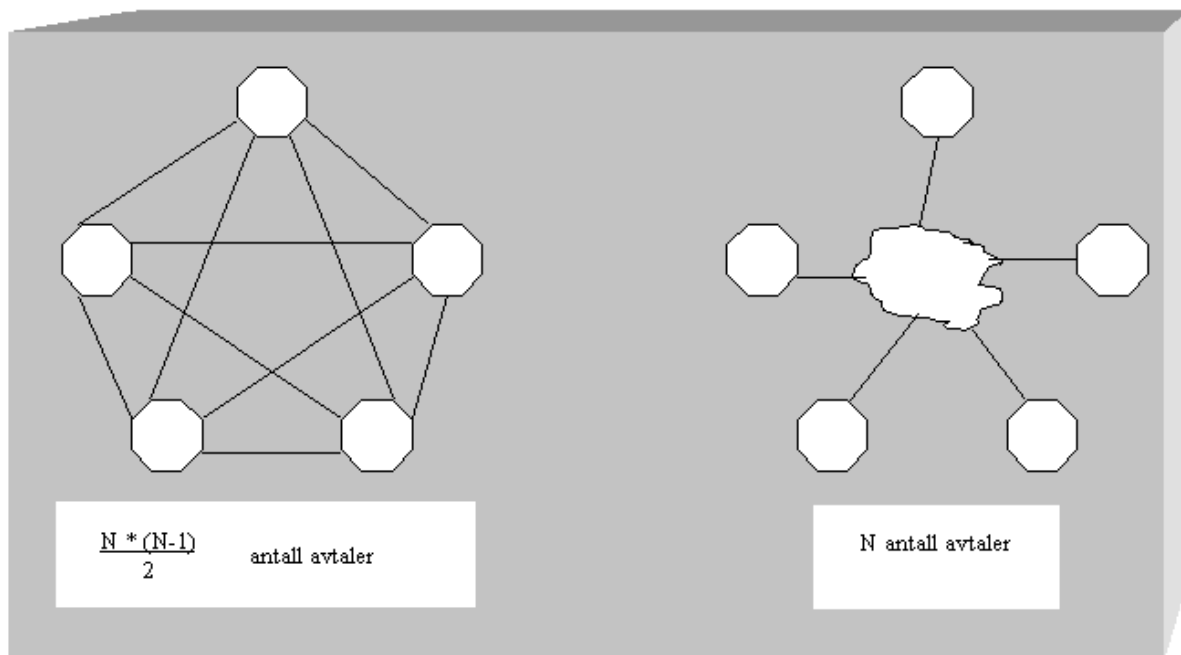
Innenfor Informasjonsteknologi får infrastruktur sin betydning blant annet når man begynner å snakke om nettverk - men også i betydningen informasjonsinfrastruktur. Når man sier informasjonsinfrastruktur favner dette videre enn valg av kommunikasjonsprotokoller. Hvordan informasjon lagres, behandles, overføres og presenteres vil være en del av dette bildet.

Å standardisere vil i dagligtale si at man skal gjøre som det andre har gjort, eller som man har blitt enige om. Hvordan står man så i forhold til å senere kunne være fleksibel overfor endringer? Det kunne være ønskelig å si "Ja takk begge deler" [Ole Brum sitat]

4.4.1 HVA GJØR EN INFRASTRUKTUR IRREVERSIBEL?

Informasjonsinfrastruktur (II) er et noe ullent begrep. Mange har sett på hvordan denne type infrastruktur må kunne endres i sin levetid ^[O Ha 1] Artikkelen illustrer godt hvordan standardisering må avbalanseres mot behovet for endring og er derfor et godt utgangspunkt for å se på hva som gjør infrastruktur irreversibel.

4.4.1.1 Hvorfor standardisere II?



Figur 29 - hvorfor standardisere?

En enkel illustrasjon på hvorfor det lønner seg å standardisere kan være antall avtaler som må inngås mellom for eksempel 5 parter som ønsker å kommunisere seg i mellom.

Med en standard kan man raskere og billigere oppnå kommunikasjon mellom flere parter. Standarder er en absolutt nødvendighet for at II skal eksistere. For å kunne kommunisere må partene kunne bruke felles standarder - dette er det felles språk, protokollen som anvendes.

4.4.2 FORHOLDET MELLOM FLEKSIBILITET OG STANDARDISERING

Standardisering vil ofte komme i situasjoner der enkelthendelser vil kreve at standardene er fleksible og de er enkle å endre [IIS-bok] Historien til Internett vil kunne tjene som et eksempel på dette. Som formulert i et internt dokument som beskriver Internetts standardiseringsprosess: "Fra starten av har Internett vært, og det forventes å fortsette å være, et system som utvikler seg og hvor deltakerne stadig forventer nye krav tilfredsstilt og ny teknologi legges inn design og dermed realiseringen "

Dette behovet for endring ser ikke ut til å stoppe. Eksemplet i så måte kan være behovet for nå å utvide adresseområdet IP. Alle noder (datamaskiner, terminaler, printere osv) har hver en unik adresse. Hvis man ikke finner en løsning vil Internetts ekspansjon stoppe, og dermed ikke lenger være et globalt nettverk. Endringen til en ny IP må foregå slik at også gammel versjon kan fortsatt å fungere ved siden av.

4.4.2.1 Hvordan muliggjøre fleksibilitet og mulighet for endringer?

Det er kjent teknikk å benytte modularisering som strategi for design. Hva angår utviklingen innenfor informatikk, herunder II, så har dette vært en systematisk strategi gjennom utviklingen av verktøy, programmeringsspråk (objektorienterte språk) og metodikk for design (objektorientering i design). Denne dekomponering og modularisering er også basis for fleksibilitet i II; fleksibilitet forutsetter modularisering. Dette blir ofte kalt "black-boxing" - ved at kun yttersiden "interface" er kjent - eller er av betydning. Innersiden betyr ikke noe og kan dermed godt endres hvis bare yttersiden holdes lik. Modularisering vil ofte føre til en lagdelt eller hierarkisk modell - OSI modellen er her et godt eksempel. Her tilbyr et lag tjenester til laget over basert på laget under, og kommunikasjonen foregår med tilsvarende lag på den andre siden. Modularisering kan også hindre kobling og overlapp mellom moduler ved å holde dem rene. Dette er tilkjennegitt når nye tjenester kan legges til uten at dette får føringer for det eksisterende. Et eksempel her er World Wide Web, hvor protokollen HTML hopper over de elementene den ikke skjønner. På denne måten kan nye og gamle implementasjoner leve side om side.

4.4.3 HVA HINDRER SÅ FLEKSIBILITET OG EVNEN TIL FORANDRING?

Det er å forstå at når modularisering ikke vedlikeholdes så vil dette hindre endringer til et II. Tre eksempler hentes fra [IIS-bok]. Et er hvordan e-mail adressen i OSI er bygget opp. Den enkeltes identitet skilles ikke fra hvordan en person blir lokalisert. Dette hindrer fleksibilitet fordi når en organisasjon knytter seg til en annen provider, så må alle brukere få nye adresser. Et annet eksempel er hvordan protokollene i OSI er bygget opp. Riktignok er det en streng hierarkisk oppbygging, men hvert lags protokoll er så pakket med funksjonalitet at det nesten umulig å implementere og enda mye vanskeligere å endre. Det er lettere å endre en enkelt komponent enn en kompleks en. Internett protokollene er mye enklere og er derfor enklere å endre. Det tredje eksemplet for hinder til fleksibilitet er utbredelsen, det vil si antallet ganger en protokoll er i bruk. Etter hvert vil det å sende ut endringer i ett vidt nettverk bli krevende. På denne måten vil nettverket kunne vokse seg ut av kontroll.

4.4.4 KONSEPTET ANVENDT I UTVIKLING AV INFORMASJONSINFRASTRUKTURER!

Det som også nevnes i artikkelen er hvordan den menneskelige prosess etter hvert kan hindre fleksibilitet. I utgangspunktet har man en åpen diskusjon om alt, nye tolkninger er mulig osv. En lukking vil etter hvert oppstå når enighet oppnås. Det vil si når de som er med bestemmer at problemer som har oppstått underveis har funnet sin løsning. Dette omtales som konsensus.

4.4.5 HVORDAN II BLIR IRREVERSIBELT

Det blir hevdet at OSI feilet fordi den er "installed base hostile" [IIS-bok] (Rose 1992; Stefferud 1992). OSI var med andre ord ikke nært nok knyttet til den allerede installerte kommunikasjonsinfrastruktur. Det sies også om Internett at dette kan være i ferd med bli irreversibelt. Dette relateres til det tidligere omtalte behovet for å endre IP. Dette skyldes ikke bare som tidligere nevnt hvordan IP er utbredt i en vid skala. Men også det faktum at mange protokoller er tett sammenkoblet. Mange andre komponenter er avhengig av IP. Dette illustreres ved at 27 av 51 av de gjeldende standarder må endres som følge av endringen [IIS-bok] (RFC 1995, 38).

I tillegg til denne tekniske bakgrunnen for irreversibilitet har også den menneskelige faktor en rolle. Etter hvert som antallet aktører, organisasjoner og institusjoner øker også antallet relasjoner mellom partene, dette fører til stor grad av irreversibilitet. Dette kan nå sees ved innføringen av nye kommersielle tjenester som tilbys av forskjellige organisasjoner. Overgangen til ny IP må koordineres mellom alle partene. Det vil lett kunne bli slik at alle venter på alle. Dette gjør det vanskelig å være den første.

4.5 IT-STRATEGI

Forsvarets Overkommando gjennomførte i perioden mai til desember 1996 prosjektet PIFO^[PIFO-96], felles prosess- og informasjonsmodell for Forsvaret. PIFOs mandat var å utvikle en felles prosess- og informasjonsmodell for Forsvaret. Prosjektet skulle også utvikle et felles rammeverk (FISPRO) for styring av alle aktiviteter i utviklingen av Forsvarets informasjonssystem (FIS). Det anbefales at FISPRO etableres som en selvstendig enhet under STSJ/FO og gis ansvar og myndighet til å styre prosjektene for videreutvikling av FIS. Jeg har ikke forespurt særlig grundig om hva de endelige resultatene av dette ble. Av konkrete resultater jeg er kjent med utkom Forsvarssjefens IS-direktiv^[IS-direk] med hjemmel i Forsvarsdepartementets retningslinjer^[FD-retn]

I 1998 ble det gjennomført et prosjekt kalt Fokus, og dette konkluderte med at på tross av disse retningslinjene så spriker det veldig i Forsvarets IT-satsinger. "FOKUS skisserte en situasjon som i stor grad er preget av prosjekter som tar frem teknisk ambisiøse løsninger uten at prosjektene styres for å realisere gevinster, og uten at prosjektledere og linjeledere i nødvendig grad opplever dette som en del av sin oppgave"

Et slikt utsagn er i tråd med ideen i PIFO, der man ser IT som et virkemiddel for å oppnå andre mål. Som en følge av at det spriker på IS/IT-siden ble IS/IT også et av delprosjektene i omstillingsprosjektet ARGUS^[ARGUS] i Forsvaret. Den gang var det ikke klart hvordan man har tenkt å se for seg IT som et virkemiddel.

Forsvaret har i perioden utarbeidet oppdaterte retningslinjer, der det særlig er IKT policy dokumentet^[IKT-POL] som er sentralt. Det viktigste her er at det nettopp er én policy for hele Forsvaret. Selv dokumentet er svært kort, og gir lite detaljert informasjon om hvordan man skal nå de målene

som er satt. Jeg vil likevel fremheve at fokus på overordnet styring så langt har vært det viktigste resultatet. Jeg har likevel forventninger til hvordan dette skal utvikle seg fremover, da policydokumentet fremhever at ”Styring av militær tilpasning og anvendelse av IKT skal gjøres gjennom bruk av arkitekturer. Fokus skal legges på å identifisere de viktigste egenskapene ved Forsvarets virksomhet, samt grensesnitt og informasjonsflyt mellom ulike deler av virksomheten”

Dette skal blant annet gjøres gjennom å fokusere på:

- IKT som katalysator for endring og gevinstrealisering,
- Brukerfokus samt
- Modularitet og standardiserte grensesnitt

Dette er momenter som også denne oppgaven her forhåpentligvis vil kunne bidra til.

4.6 ANSKAFFESESREGIME

Her forsøker jeg svært kort å angi at det eksisterer et sett med regler og metoder for anskaffelse i Forsvaret. Det er ikke meningen å gå i detalj, men det illustrerer at på enkelte områder har en stor organisasjon som Forsvaret et etablert sett for metoder og til dels verktøy.

4.6.1 ANSKAFFESESREGELVERK FOR FORSVARET

Anskaffelsesregelverk for Forsvaret ^{ARF} erstatter fra 1. september 2004 de tidligere Bestemmelser vedrørende anskaffelser til Forsvaret. ARF bygger på et oppdatert konsept for fremskaffelse av materielle kapasiteter i forsvarssektoren ^{KONSIF}. Det nye konseptet benytter gjennomgående begrepet fremskaffelse istedenfor anskaffelse/investering. Dette er gjort med hensikt, for å fokusere på at forsvarssektoren ikke lenger må kjøpe og eie alt materiell og alle bygg og anlegg selv, men i stedet systematisk skal vurdere alternativer til tradisjonelle innkjøp før man tar beslutning om valg av løsning

ARF beskriver fasene av en anskaffelse etter at beslutning er gjort om å iverksette en anskaffelse. Her beskriver videre hvordan gjennomføring og kontroll skal utøves.

Det er videre slikt at alle anskaffelser i Staten skal forholdes seg til Lov om Offentlige Anskaffelser. ^[LOA] Siden dette primært er en oppgave innenfor systemutvikling og informatikk går jeg ikke nærmere inn på dette. Det er likevel verdt å bemerke at Forsvaret har egne muligheter og regler gjennom ARF. Blant annet har Forsvaret mulighet til å avgrense konkurranse dersom det anses nødvendig, se for øvrig [ARF].

4.6.2 PRINSIX

Forsvarets modell for prosjektstyring har i mange år vært uforandret. Etter St.prp. nr. 1 (2005-2006) ^[ST-PRP1]: ”refereres det til at ... det er iverksatt et nytt konsept for investeringsstyring. Det er særlig lagt vekt på å styrke helhetlig og strategisk styring både med hensyn til nyanskaffelser og erstatning av vedtatte kapasiteter. Arbeidet med detaljert avklaring av roller og ansvar, oppdatering av Forsvarets prosjektstyringsrutiner (PRINSIX) samt verktøy for porteføljestyring, har fortsatt i 2005.” Dette må sees i sammenheng med ”Nytt konsept for investeringsstyring i Forsvaret”, ^[INVSTY]

Så langt har dette i liten grad endret praksis. I dagens FLO er det prinsippene i PRINSIX ^[PRINSIX] som fortsatt ligger til grunn.

PRINSIX er i hovedsak et dokumentsentrisk og milepælsdrevet prosjektstyringssystem. Det er i den senere tid blitt et mer profesjonelt utdanningsopplegg, der utdanningsopplegg er samkjørt mot norske standarder for prosjektledelse og selve gjennomføring av utdanninger er satt ut fra Forsvaret.

4.6.3 VIDERE UTVIKLING

Dette er eksempler på eksisterende regler og metoder. Disse regler og metoder er i seg selv i stadig utvikling. I denne oppgaven ser jeg på sammenhengen mellom anskaffelse, prosjektgjennomføring og systemutvikling.

Endringer innenfor et område bør kunne gjennomføres uten at dette får store følger for andre områder.

5 ET RAMMEVERK FOR SYSTEMUTVIKLING

Her presenteres et rammeverk for systemutvikling, modellen baserer seg på MACCIS, et produkt fra SINTEF Tele & Data[MACCIS]. I første kapittel beskrives hva et arkitekturrammeverk er mens kapittel 2 beskriver det konkrete rammeverket MACCIS.

5.1 ET ARKITEKTURRAMMEVERK

SINTEF Tele og Data produserte på oppdrag fra Hærens Forsyningskommando Minimal Architecture for CCIS – MACCIS. MACCIS ble i første versjon levert 1999-03-04, denne presentasjonen bygger på en oppdatert versjon 2 se ^[MACCIS].

Motivasjonen for å lage MACCIS har vært nettopp de krav som er framført i kapittel 3, behovet for et fleksibelt K2IS som er åpen for endringer etter hvert som kravene endres, og behovet for et K2IS som er interoperabelt over tid med andre nasjonale systemer, så vel som NATO- systemer.

For å forstå oppbyggingen kreves det at noen termer gjøres kjent

5.1.1 TERMER OG BEGREPER

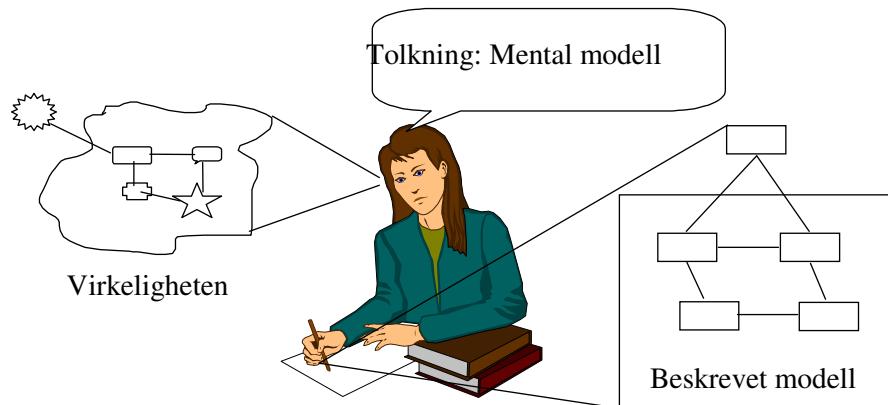
Denne delen beskriver og definerer de termer og begreper som benyttes. Det er avgjørende at disse termer og begreper forstås likt for at vi skal kunne bruke de i systemutvikling.

De viktige termene er

- Modeller
- Systemnivå
- Ståsted, synsvinkel og syn
- Raffinering
- Komponenter
- Objekter

5.1.1.1 Modeller

Når vi skal lage en modell er det avgjørende å forstå hva modellen skal illustrere. Mange egenskaper vil være skjult i en modell – faktisk er det hele grunnlaget for at vi ønsker å lage en modell – nemlig å abstrahere vekk alle unødige detaljer. Når vi abstraherer, kan vi gjøre dette på flere måter. Vi kan velge å abstrahere langs flere retninger, det vil si vi kan utelate deler av informasjonen avhengig av hva som er hensikten med modellen. For eksempel, kan vi bare se på prosesser og aktiviteter på den ene siden, eller vi kan velge å se på informasjonen som skal håndteres og hvordan denne er sammensatt. Her vil vi presentere 3 dimensjoner som er grunnlaget for forståelsen av MACCIS. Først er det dog viktig å forstå hva som menes med ”modell”. Modell brukes som term for å spesifisere en enhet i den virkelige verden. På denne måten er modeller spesifikasjoner på et spesielt format (typisk i et modelleringsspråk som UML) – hvor alle symboler i modelleringsspråket har en definert og felles tolkning.



Figur 30 Modelleringsprosessen: Skapelse av en modell av virkeligheten.

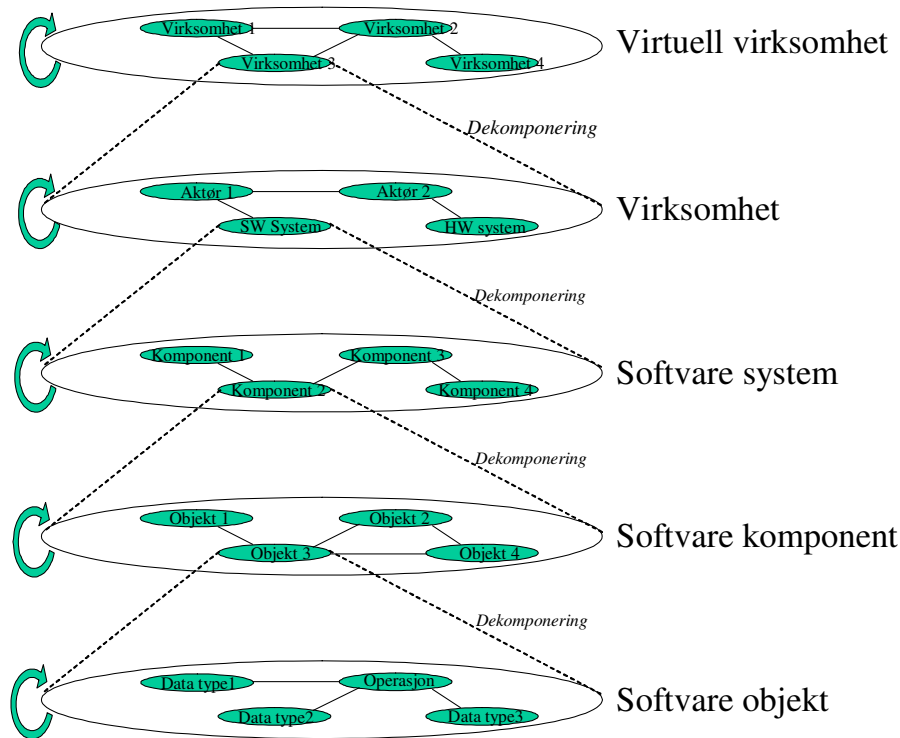
Modeller er bilder på virkeligheten, og er aldri komplette. Hver modell fremhever enkelt trekk som vi ønsker å modellere. Den verden som er av interesse for systemet er i andre sammenhenger kalt Anvendelsesområde ^{[OBJ-AN93]²⁰} I systemutviklingssammenheng ved Institutt for Informatikk, for eksempel i faget IN-260, har det tidligere blitt anbefalt at for å lage modeller til dette formålet kan man bruke rike bilder. I denne oppgaven hevdes det at siden modellen av virkeligheten også skal brukes til å spesifisere rammene til systemet, ønsker vi å bruke mer formelle modeller. Her vil rike bilder ikke være tilstrekkelige alene, vi trenger modeller som er mer entydige. Utbredelsen av Universal Modeling Language ^[UML] er i dag et tegn på at modellering har en fått en vid anvendelse. Modellering har en lang tradisjon som del av systemutvikling. Jeg går ikke inn på de historiske sidene som ligger bak utviklingen av MACCIS. Det er lang tradisjon for at den del av virkeligheten som systemet skal bidra til å understøtte – problemområdet, har blitt modellert. Her argumenteres det i tillegg for at den virksomheten som systemet skal inngå i kan modelleres. For eksempler og ytterligere motivasjon for modellering henvises det til MACCIS.

5.1.1.2 Systemnivå

Når vi modellerer har vi ofte "system" i tankene. "System" i denne sammenhengen blir brukt på mange nivå – fra et høyt virksomhetsnivå hvor deler av virksomheten samvirker som et system, til software objekter hvor data typer og operasjoner på disse utgjør et system. Disse forskjellige nivåer av system har klare relasjoner. En virtuell virksomhet består av samhandlende virksomheter, en virksomhet består av samhandlende aktører og tekniske systemer. Et softwaresystem som eksempel på et teknisk system består av komponenter, en software komponent består av samhandlende software objekter, og til slutt består et software objekt av datatyper og operasjoner på disse. Disse nivåene, selv om de langt på vei er godt innarbeidede begreper, er bare eksempler på nivå – det er ikke de eneste nivå som kan tenkes. Faktisk kan man ha et utall forskjellige nivå ved å benytte rekursjon. For eksempel kan en virksomhet igjen bestå av virksomheter, og et teknisk system kan igjen bestå av

²⁰ Anvendelsesområde: En eller flere organisatoriske enheters administrasjon, styring eller overvåkning av et problemområde [OBJ-AN93](I MACCIS og ellers i amerikansk litteratur kalles anvendelsesområdet for "universe of discourse" -UoD.

sammensatte tekniske system. Figur 31 Systemnivå viser de flest tenkelige nivå. Rekursjon er illustrert på hvert nivå ved den lille grisehalen som peker tilbake på det samme nivået.



Figur 31 Systemnivå

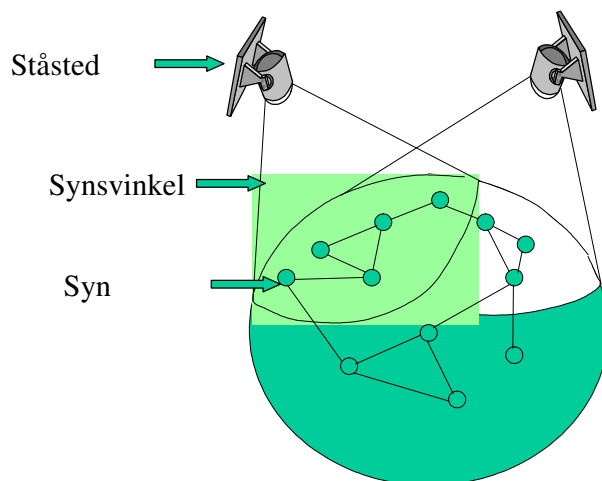
En modell skal enten være en modell av enheter på et nivå og hvordan disse samhandler eller hører sammen, eller det skal være en modell av enheter på et nivå og hvordan de samhandler eller hører til enheter på et annet nivå. Dette siste kaller vi dekomponering ettersom det illustrer hvordan en enhet er satt sammen av komponenter fra et annet systemnivå. Eksempler på dette kjenner vi fra organisasjoner, der en softwarebedrift består av menneskelige aktører som for eksempel, sjef, sekretær, utviklere osv, samt tekniske systemer som bygninger, datamaskiner, stoler, software utviklingsmiljø osv. Det er en forskjell mellom dekomponering slik vi benytter det for å traversere systemnivå og relasjonen "inneholder" som brukes til å modellere medlemskap. Der dekomponering modellerer relasjonen "består av", så modellerer relasjonen "inneholder" bare at den inneholder. Slik inneholder en bygning stoler, men en bygning består ikke av stoler – derfor kan vi ikke dekomponere en bygning til stoler og andre ting. En bygning derimot består derimot av vegger, gulv etc. – slik kan vi dekomponere bygningen i nettopp vegger, gulv og tak.

For software systemutvikling er virksomhetsnivå og software systemnivå to viktige systemnivå. Virksomhetsnivå inneholder menneskelige aktører og tekniske systemer, så som software system. Modeller av virksomheten med software system sier oss noe om hvordan brukerne vil bruke software systemet. Med andre ord vil en virksomhetsmodell som også omhandler software system, gi de funksjonelle krav til dette systemet. Det er derfor en virksomhetsmodell er helt avgjørende for systemutviklingen for å sikre at virksomhetens behov er ivaretatt i systemet. Hvis derimot virksomhetsmodellen ikke er koblet til systemnivået vil de miste mye av sin verdi. Software systemet i virksomhetsmodellen må minst dekomponeres til et nivå hvor komponentene som samhandler trer frem slik at kravene til systemet blir ivaretatt av en gitt konfigurasjon. Egenskapene til et system slik

det er beskrevet på virksamhetsnivå må løses ved at de samme egenskaper ivaretas på systemnivå ved en gitt konfigurasjon.

5.1.1.3 Ståsted, synsvinkel og syn

Et system på et hvilket som helst nivå kan sees fra forskjellige ståsted. Motivasjonen for ståsted kommer av at en komplett spesifikasjon av ethvert ikke trivielt system vil inneholde svært mye informasjon. Denne informasjonen kan deles inn ved hjelp av ståsted og synsvinkel for å ivareta forskjellige hensyn for hver enkelt interessent. En synsvinkel er ikke noe annet enn en abstraksjon som beskriver en spesifikasjon av hele systemet. En synsvinkel er en projeksjon mot et sett med hensyn, og må ikke forveksles med systemnivå som ble beskrevet over. Synsvinklene må være konsistente seg i mellom, da synsvinklene bare er forskjellige perspektiver på det samme systemet. Spesifikasjoner fra flere ståsted kan delvis ha overlapp, og kan også inneholde forskjellige aspekter av det samme systemet – det er derfor helt avgjørende at synsvinklene seg i mellom er konsistente. Figur 3 illustrerer metaforen ståsted, hvor enkelte aspekter ved et system trer frem når de er belyst fra forskjellige ståsted. Ståsted og perspektiv betegner hvordan vi ser på systemet, mens syn er hva vi ser når vi velger en synsvinkel fra et ståsted. Et syn vil typisk bli representert ved en eller flere modeller.

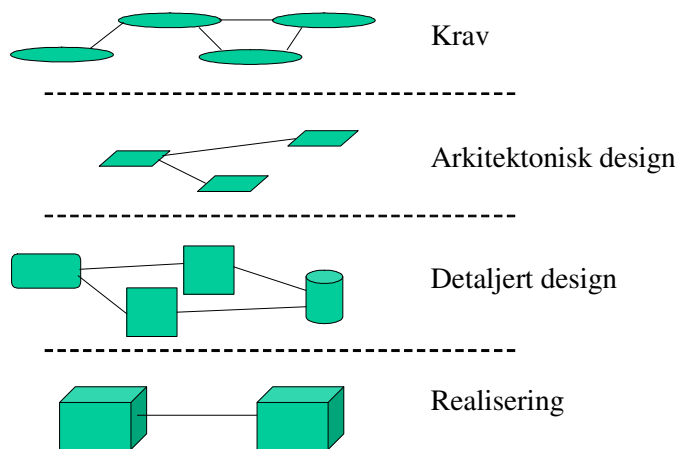


Figur 32 Ståsted – Perspektiv - Syn

5.1.1.4 Raffinering

Raffinering er enda en måte å abstrahere på. En vanlig måte å forstå raffinering er å legge til mer detaljer. Et eksempel kan være en person har en bil. En raffinering her vil være at personen har en hvit bil. Vi forstår raffineringen, der den første påstanden er sann, men på det neste nivået gir oss flere detaljer. Vi kan altså si at raffinering er det å klargjøre uspesifiserte detaljer ved den første påstanden.

Ved systemutvikling er raffinering en avgjørende aktivitet. Det er vanlig å starte med en underspesifisert modell, det vil si en modell som har noen uspesifiserte men relevante aspekter. Denne modellen raffineres ved å stadig redusere de uspesifiserte aspekter til vi har et grunnlag for realisering. Typisk brukes en trinnvis tilnærming for å raffinere, der noen vanlige nivåer er: Krav, arkitektonisk design, detaljert design og realisering.



Figur 33 Raffinering

Dekomponering har vi tidligere sett som en form for strukturell raffinering. Når vi går ned et systemnivå vil flere detaljer bli spesifisert ved å tilkjenne systemets komponenter og hvordan disse samhandler. Raffinering av oppførsel er en annen type raffinering som gir detaljer som avgrenser handlingsrommet til systemet. Eksempelvis, spesifikasjon av oppførsel til en bil kan beskrive at hastigheten skal endres når man trykker på gasspedalen når bilen er i gir og motoren er i gang. En raffinert beskrivelse av det samme kan fortelle at hastigheten skal øke i dette tilfellet. Dette reduserer oppførselen til bilen med det samme stimuli (gasspedalen presses ned). Vi har tidligere påpekt sammenhengen mellom modeller på forskjellige systemnivå. En slik raffinering av oppførsel er et eksempel på denne sammenhengen mellom krav på virksomhet og løsninger på systemnivå. Det har ingen hensikt å lage et softwaresystem som ikke passer inn i virksomheten.

5.1.1.5 Komponenter

MACCIS som modell, metode og teknikk fremmer objektorientert analyse og design. I objektorientert analyse[OBJ-AN93] brukes objekter til å beskrive fenomener i informasjonssystemets problemområde. Objektorientert analyse er kjennetegnet ved at vi starter med å se på det som informasjonssystemet skal handle om. Ved andre metoder som for eksempel strukturert analyse, som er en funksjonsorientert metode, ville vi straks fokusere på kravene. Hvis vi utvikler etter en funksjonsorientert metode, setter vi straks i gang med å beskrive ved hjelp av dataflydiagrammer og prosessbeskrivelser hvordan blanketter beveger seg i problemområde, sammen med datafly i Edb-systemet. Vi interesserer oss derimot ikke for de ansatte og deres gjøren og laden, men kun for de data som de behandler og handlinger som de utfører. Objektorientert analyse er slik en utvidelse i forhold til tradisjonelle funksjonsorienterte metoder. Fordelen er en bedre begrepsmessig forståelse, ved at vi ikke ene og alene ser problemområdet gjennom det filter som dagens praksis er i virksomheten. Et forbedret begrepsmessig grunnlag vil gi et informasjonssystem som passer til problemområde og som blir lettere å vedlikeholde og videreutvikle.

Dette setter fokus på å bruke konsepter og komponenter fra virksomheten gjennom analyse og design så vel som i implementasjonsfasen.

MACCIS beskriver utvikling av objektmodeller som er uavhengig av anvendelse og realisering i software-komponenter. Dette er ulikt tradisjonelle analyse og design metoder ved at vi i MACCIS skaper og vedlikeholder en binding mellom anvendelsesområdet og problemområdet til softwaren. På denne måten reflekterer en software komponent policy og virksomhetsstrategien.

En av MACCIS sine grunnleggende egenskaper er nettopp at informasjonssystemet kan og skal være sammensatt av komponenter som speiler anvendelsesområdet og problemområdet.

Hva er så en komponent? Komponenter i følge MACCIS er først objektorientert tankegang og så et skritt til. Forutsetningen for dette er en grunnleggende egenskap ved objekter. Komponenter baserer seg på at også objekter er satt sammen av objekter!

Et eksempel kan her være et Geografiske Informasjonssystem. Hvis du utvikler et større software objekt som skal håndtere visning av et geografisk kart i et vindu på skjermen, så vil du raskt se at dette er noe som også vil være nyttig i andre programmer. I stedet for å skrive koden til et program selv, kan du kanskje velge fra objekter som andre har laget – og kanskje laget godt også. Dette er en fristende tanke hvis målet er å lage et stort system. Det er dog ikke bare å ta ethvert objekt å gjenbruke dette i ditt program direkte. For at objekter skal kunne gjenbrukes i andre program, må det være beskrevet på en slik måte at du senere kan avgjøre hva det gjør og hva hvordan du kan bruke det. Komponenter er hva vi kaller en slik løsning. Ved å pakke objektet inn slik at i tillegg til koden beskrives også denne på en form som andre kan lese. Hvis denne pakken inneholder en standard måte eller protokoll for å få tilgang til operasjonen i objektet får du en komponent. De ekstra data om objektets data, kalt metadata, gjør det mulig å fortelle hva komponenten gjør, operasjonene tillater deg å bruke komponenten riktig.

For at et objekt skal være en komponent må det være i seg selv være selvstendig eksekverbart og ha et veldefinert grensesnitt. En komponent skal være gjenbrukbar, og den kan bli solgt eller kjøpt som hylleware. Slik kan f.eks. en knapperad eller en annen del av brukergrensesnittet være en komponent, hvis det har et veldefinert grensesnitt.

Komponenter fremmer gjenbruk av kode Ved hjelp av komponenter kan du plukke biter av software fra hyllen og plugge de rett inn i ditt program. Slik kan du spare enormt med tid, ressurser og svette over selvkomponert kode spekket med feil.

Komponentens veldefinerte grensesnitt kalles ofte for en komponentkontrakt. Denne beskriver komponentens protokoll. Kontrakten må kunne benytte komponenten på en sikker måte, og skal minst inneholde beskrivelse av metoder og semantikken til komponenten. I MACCIS refereres det til tilgjengelige komponentteknologier i dag.

5.1.1.6 Objekter

Objekter representerer en formålstjenlig måte å dele anvendelsesområdet og problemområdet inn i deler som har sin egen informasjon og som har tilstrekkelig med tjenester til å utføre det som forventes av andre. Objekter har mange egenskaper:

Objekter øker vedlikeholdbarheten ved å innkapsle den informasjon som representeres i hvert objekt

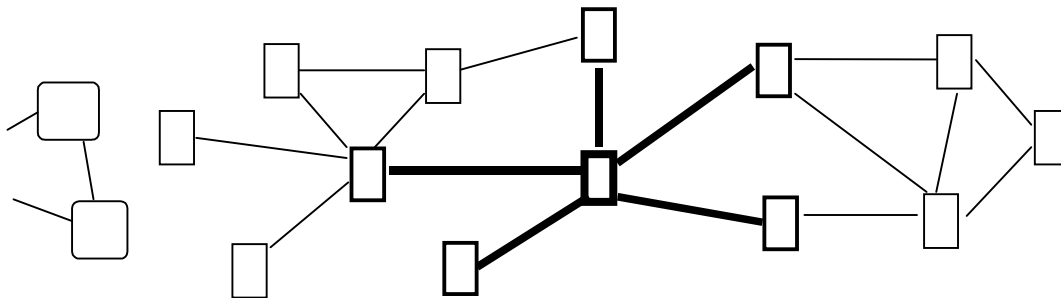
Informasjon oppbevares og håndteres på ett sted. Dette gjør det enklere å forstå et kompleksiteten ved at informasjon er plassert sammen med den funksjonalitet som har tilgang til den.

Kontroll til informasjonen kan styres. Andre objekter kan ikke endre data på andre måter enn den det kontrollerende objektet tillater.

Objekter som konsept er lett å forstå, ettersom de mimer objekter i den virkelige verden uten at vi trenger å bry oss hvordan med hvordan de ser ut inni.

Det er viktig å forstå at å bruke objekter er en måte å tenke på. Termen objekter dukker opp i svært mange sammenhenger, uten at ytterligere eksempler tas opp her, se for øvrig[MACCIS].

Figur 34 Objekter i samspill, forsøker i en enkel illustrasjon å vise en modell satt sammen av objekter.



Figur 34 Objekter i samspill

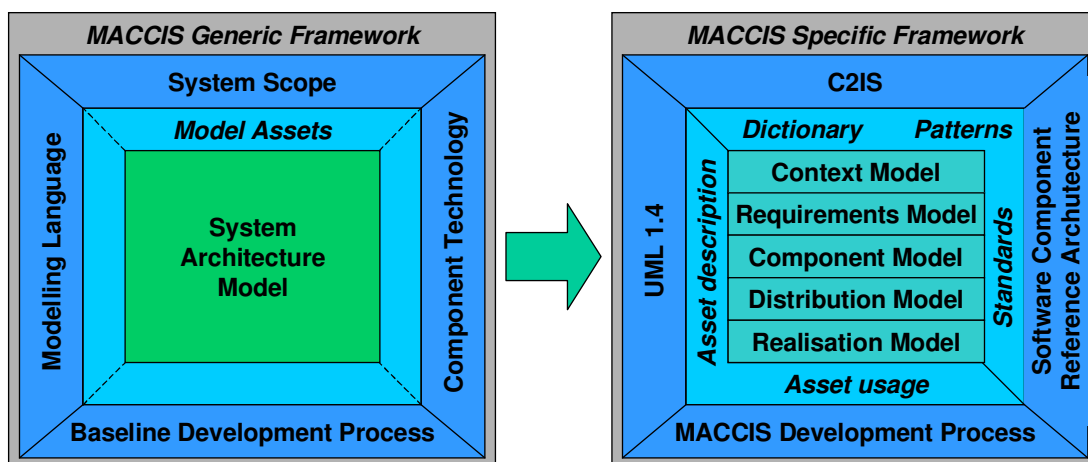
I et objektorientert verdensbilde, består verden av objekter i samspill hvor hvert objekt har et veldefinert og begrenset ansvar, kjenner sine naboer, vedlikeholder sin egen tilstand og sitt eget bilde av verden er robust og kjenner ikke til alt alene!

5.1.2 ET RAMMEVERK FOR ARKITEKTURBESKRIVELSE

5.1.2.1 Innledning

Her gis en oversikt over arkitekturrammeverket. MACCIS beskriver et generisk rammeverk for systemarkitektur som egentlig kan brukes også i andre domener.

Figur 35 MACCIS Arkitekturrammeverk viser både det generiske og den spesielle K2IS versjon av rammeverket.



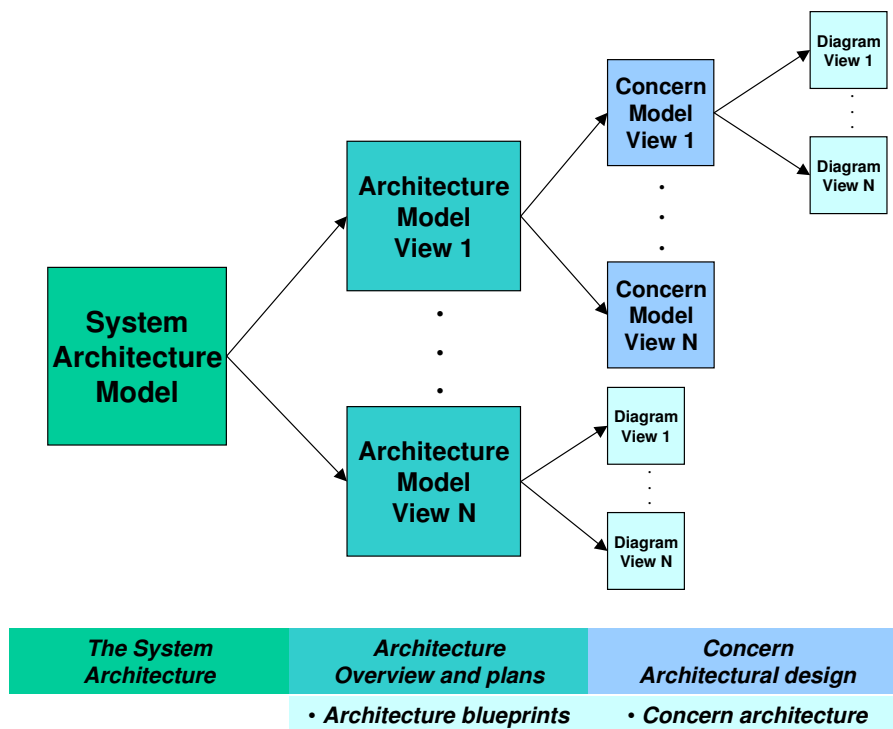
Figur 35 MACCIS Arkitekturrammeverk

Det generiske rammeverket fokuserer på systemarkitekturmodeller. Systemarkitekturmodeller beskrives i et modelleringsspråk etter en systemutviklingsmetode. Systemet har et domene som beskrevet i systemfokus og et sett målteknologier. For å understøtte modellering av systemarkitekturen er det tilgjengelig en del ressurser tilpasset språk, prosess, fokus og teknologi.

Den spesifikke arkitekturen viser det mer raffinerte rammeverket for K2IS. Ved å vise det generiske rammeverket er det lettere å tilpasse endringer som kommer som en følge av ytre påvirkninger på rammeverket. De enkelte delene i det raffinerte rammeverket beskrives ytterligere i de følgende kapitlene.

5.1.2.2 Systemarkitekturmodell

Figur 36 viser Det grunnleggende modellhierarki som MACCIS bygger på.



Figur 36 Det grunnleggende modellhierarki

Systemarkitekturmodellen er arkitekturbeskrivelsen av det vi ønsker å kalle "systemet". Systemarkitekturmodellen kan deles opp i flere syns (basert på synsvinkler på systemarkitekturen) som beskriver viktige aspekter ved systemet vi skal se på. Innenfor hvert syn beskriver vi overordnet struktur og oppførsel ved å beskrive systemets anliggende. Hovedanliggende er systemets oppførsel. Et systems anliggende er fullt beskrevet ved hjelp av et sett av modeller – en modell for hvert perspektiv.

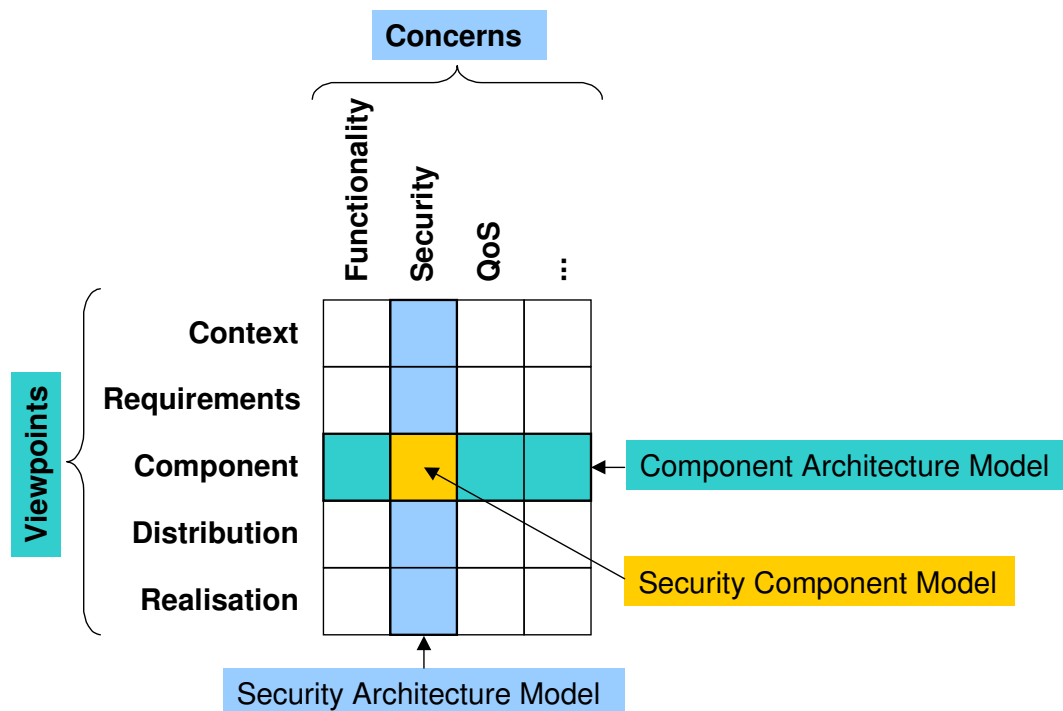
I tillegg kan vi også lage diagrammer som beskriver utvalgte deler av en arkitekturmodell, der dette er ønsket. Diagrammer er en visuell hjelp for å visualisere modellen. MACCIS anbefaler en serie av nyttige diagrammer

MACCIS definerer systemarkitektur ved hjelp av 5 arkitekturmodellssyn:

- **Omgivelsesmodell** Hensikten med omgivelsesmodellen er å beskrive virkemåten til systemet i en gitt omgivelse. Denne modellen definerer sammenhengen mellom systemets deler og omgivelsene.
- **Kravmodell** Hensikten med kravmodellen er å samle kraven til systemet

- **Komponentmodell** Hensikten med komponentmodellen er å beskrive tjenester, informasjon, komponenter og samhandling i systemet
- **Distribusjonsmodell** Hensikten med distribusjonsmodellen er å beskrive distribusjonsanliggender i systemet
- **Realiseringsmodell** Hensikten med realiseringsmodellen er å beskrive realiseringen av systemet ved hjelp av teknologi og delsystemer

Figur 37 MACCIS modellhierarki illustrerer ved hjelp av en matrise arkitektursyn og den enkelte anliggende som modellene i MACCIS adresserer. Figuren illustrerer bruk av modellsyn for komponentarkitektursynet og sikkerhetsanliggendesynet. Komponentarkitekturmodellen beskriver tjenester, informasjon, komponenter og samhandling i systemet fra for alle anliggender. Sikkerhetsarkitekturmodellen består av alle sikkerhetsmodeller som til sammen utgjør sikkerhetsarkitekturen. Komponentmodell sikkerhet beskriver sikkerhetsanliggende innenfor synsvinkel komponentarkitektur.



Figur 37 MACCIS modellhierarki

5.1.2.3 Modelleringsressurser

Modelleringsressurser er kilder som kan brukes når vi skal lage arkitekturmodeller. Eksempler på ressurser er:

- **Ordbok** Ordboken er en referanseliste over de konsepter som skal brukes når vi lager en modell, sammen med forklaring og beskrivelse av betydning og bruk.

- **Standarder** En standard er en formalisert modell eller et eksempel som er laget av et standardiseringsorgan eller på annen måte et produkt av konsensus. Når vi skal lage et K2IS vil sannsynligvis mange standarder bli tatt i bruk, og disse bør refereres eller dokumenteres.
- **Mønstre** Mønstre er en beskrivelse av et gjentakende velkjent problem med en forslått løsning. Mønstre identifiseres og kan gjenbrukes på mange systemnivå.

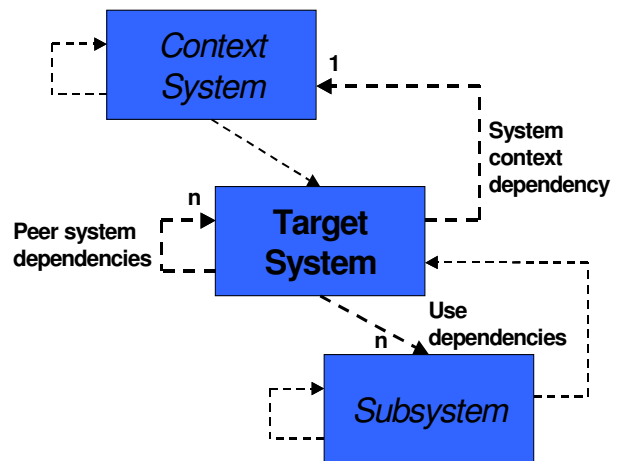
5.1.2.4 System skoping

Når vi skal beskrive arkitekturen til et system, trenger vi også å beskrive systemets omgivelser.

Videre må vi beskrive realiseringen av systemet ved å vise til systemets subsystemer.

Sammenhengen mellom disse forhold er beskrevet i Figur 38, som viser det generelle hierarki og de sentrale avhengigheter mellom byggesteinene i systemutviklingen.

Systemet som skal lages er målsystemet. Omgivelsessystemet er det høyere ordens system som målsystemet skal understøtte eller integreres inn i. Dette er vist som en "system omgivelser avhengighet". Subsystem representerer de byggeklossene som vi bruker til å realisere målsystemet. Dette er representert med "braker relasjonen". Målsystemet kan også ha avhengigheter til andre systemer på samme nivå. Dette er illustrert ved relasjonen "peer system dependencies".



Figur 38 Systemhierarki

Denne modellen er generisk og kan rekursivt benyttes slik det er beskrevet tidligere til å håndtere et uendelig antall systemer på forskjellige nivå.

Rammeverket MACCIS beskriver hvordan hvert slikt nivå kan beskrives ved hjelp av UML modeller. Rammeverket har hatt som målsetting å gjøre dette med et minimum antall modeller.

Systemmodellen brukes til å beskrive målsystemet. Figuren viser at systemmodellen er knyttet til omgivelsene gjennom den høyere ordens modell, og realiseringsmodellen knytter målsystemet til de lavere ordens systemmodeller. De enkelte arkitekturmodeller beskriver dette nærmere.

5.1.2.5 Komponentteknologi

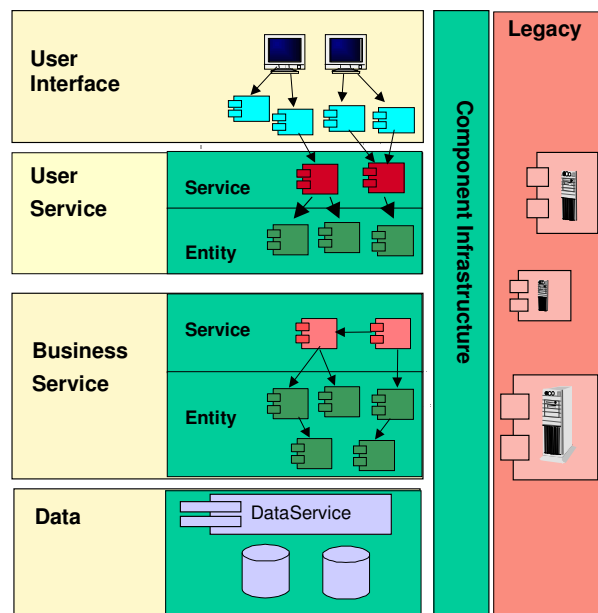
MACCIS er bygget på idéen om komponenter og objekter. Dette innebærer at den tekniske arkitekturen gjenspeiler moderne software arkitektur, basert på komponent- og objektteknologi.

MACCIS software komponentarkitektur er basert på komponentarkitekturen definert i COMBINE-prosjektet²¹.

²¹"The Open Group, Western Geco, Open-IT, Softteam, IONA Development, SINTEF, and INESC, "http://www.opengroup.org/combine/," , 2000-2002"

Komponentarkitektur bygger på en grunnleggende generisk systemarkitektur, et mønster gjerne kalt en referansearkitektur.

Denne referansearkitekturen beskriver logiske lag og rekker som hver består av komponenter. Denne lagdelingen gir oss en måte å strukturere komponentene på en fleksibel måte. Hvert lag inneholder forskjellige typer komponenter med visse egenskaper og hvert lag har sine spesifikke ansvar innenfor et distribuert system. På denne måten ivaretas distribusjon på en fleksibel måte.



Figur 39 MACCIS Referansearkitektur for Software komponenter

- **Brukergrensesnittlaget** Dette laget håndterer samhandling med brukeren gjennom et sett av brukergrensesnittskomponenter. Laget kommuniserer med brukertjenestelaget.
- **Brukertjenestelaget** Dette laget håndterer brukerens modell, det vil si håndtering av brukerens rettigheter, muligheter og informasjon i systemet. Dette er en abstraksjon av tjenester som finnes i tjenestelaget, slik at tjenestelaget blir transparent for brukergrensesnittlaget.
- **Tjenestelaget** Dette laget håndterer tjenestekomponenter med virksomhetens vidtrekkende kjernetjenester. Ettersom dette laget også håndterer virksomhetskritiske tjenester, må det også her beskytte disse tjenestene.
- **Datalaget** Dette laget håndterer datatjeneste komponenter som gir tilgang til å oppdatere dataressurser. Datalagskomponenter håndterer persistent lagring

5.1.2.6 Modelleringsspråk

MACCIS er beskrevet i språket UML. Hovedelementene herfra er: (på engelsk)

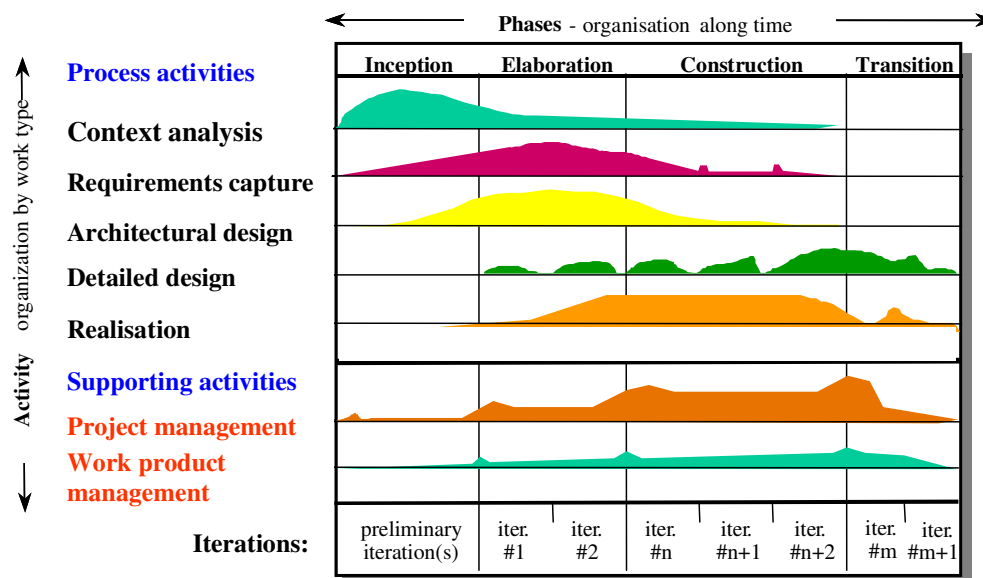
- Action states, Sub activity states and Activity diagrams

- Actors, Use cases and Use case diagrams
- Sequence diagrams / Collaboration diagrams
- Packages, Subsystems, Classes, Interfaces and Class diagrams
- Components, Nodes and Deployment diagrams

5.1.2.7 Grunnlag for en systemutviklingsprosess

Systemutviklingsprosessen er illustrert i Figur 40 MACCIS systemutviklingsprosess.

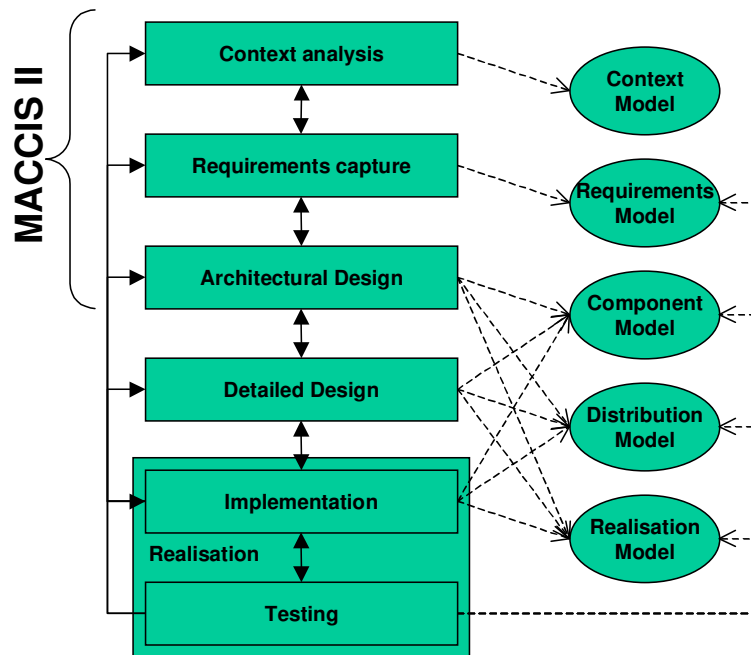
Denne figuren viser de hovedaktiviteter som gjennomføres i en systemutviklingsprosess. Fasene i systemutviklingen er vist horisontalt, mens aktivitetene er vist vertikalt. Langs hver linje representerer størrelsen på arealet den innsats som trengs for å lage de forskjellige modellene i de enkelte fasene.



Figur 40 MACCIS systemutviklingsprosess

I hver arbeidsprosess utvikles det ulike arkitekturmodeller og Figur 41 viser MACCIS utviklingsprosesser og tilhørende modeller.. MACCIS er et arkitekturrammeverk som først og fremst håndterer de tre øverste aktiviteten, analyse av omgivelser, kravinnhenting og arkitektoniske design.

Når dette er sagt vil MACCIS systemutviklingsprosess også støtte en del av de mer detaljorienterte aktiviteter som utføres mot implementasjon. Dette vises ved at der analyse av omgivelser og krav kun fører til en modell, vil arkitektonisk design bidra til flere modeller, også rettet mot realisering.



Figur 41 MACCIS utviklingsprosesser og tilhørende modeller

Systemutviklingsprosessen presenteres nærmere i kapittel 6

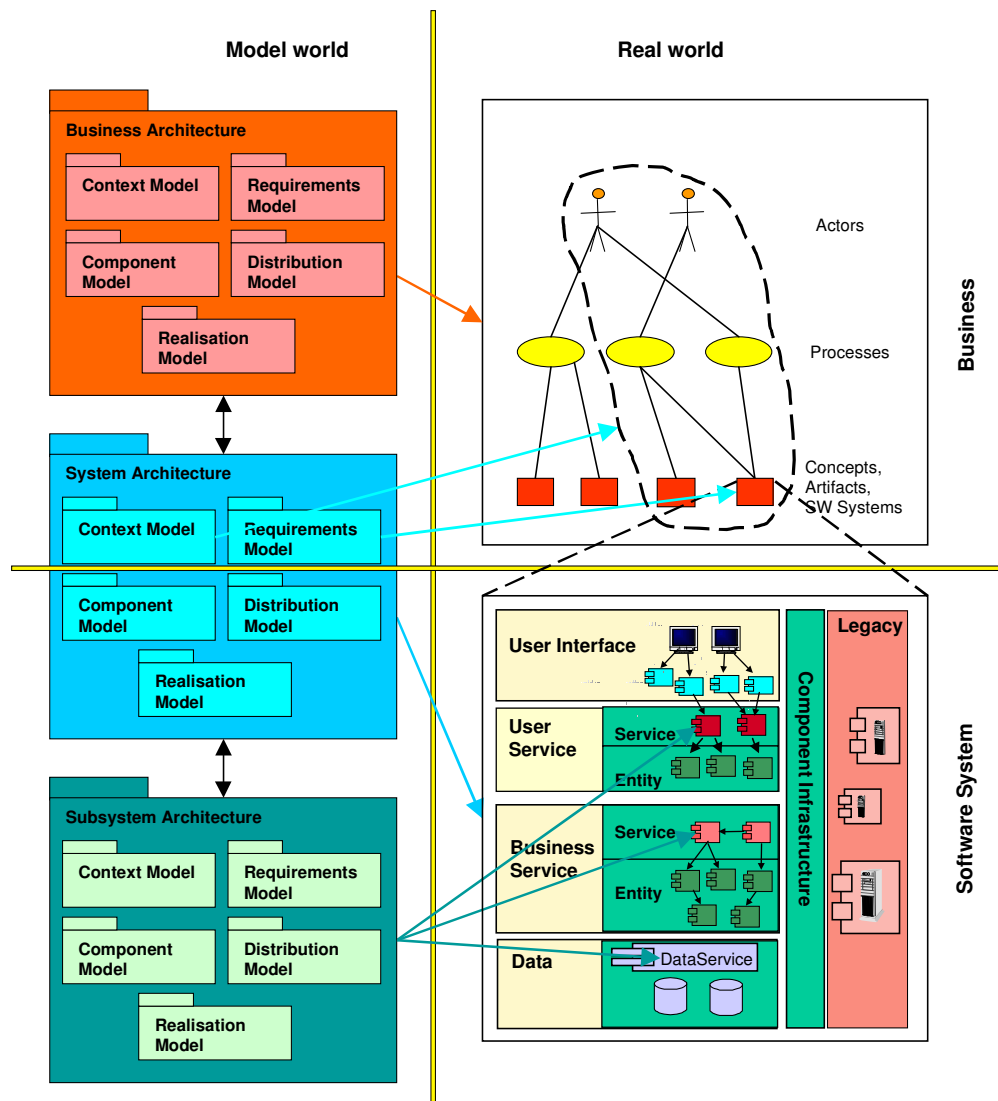
5.1.3 EN INNFORING I MACCIS

5.1.3.1 Motivasjon

MACCIS ble laget for å gjøre arkitekturinnføring i Forsvaret enklere. Første versjon av MACCIS var for mange vanskelig å ta i bruk. I versjon 2 ble dette forsøkt kompensert ved at MACCIS dokumentasjonen ble delt i to – MACCIS håndbok og MACCIS spesifisering. Arkitekturmodellene er noe forskjellig presentert i de to. Her følger en kort innføring basert på MACCIS håndbok som skal gi en god oversikt over MACCIS modellene, hvilket formål de har og hvordan de kan være til nytte i systemutviklingen.

5.1.3.2 Oversikt

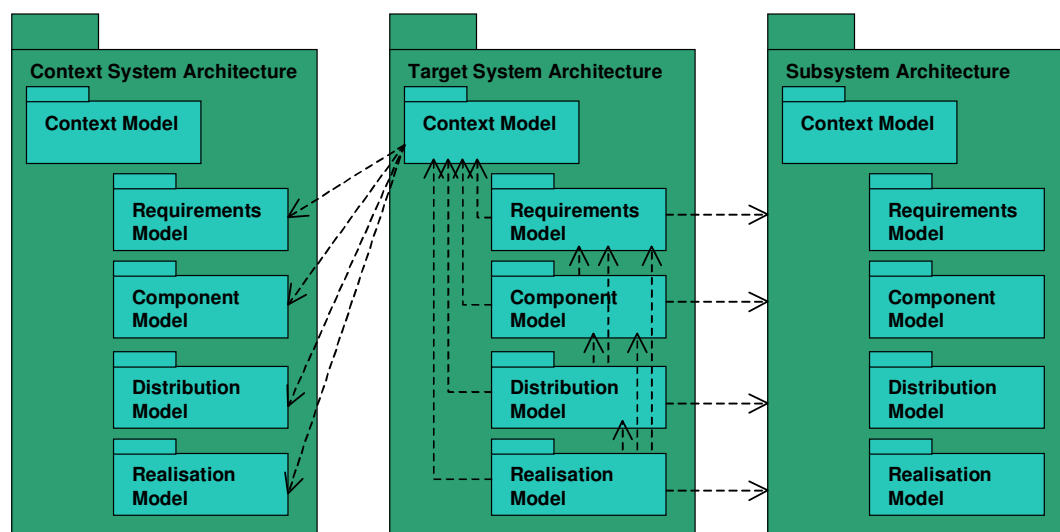
Figur 42 viser en Oversikt over MACCIS modellene og hvordan dette rammeverket kan brukes. Modellverden inneholder som vi nå har sett i kapittel 5.1.2 de fem arkitekturmodellene, som benyttes til både å beskrive virksomheten og software systemet.



Figur 42 Oversikt over MACCIS modellene

Vi har tidligere sett at målsystemet vårt avhenger av omgivelsene – virksomheten, og subsystemene er byggesteinene for arkitekturen til målsystemet.

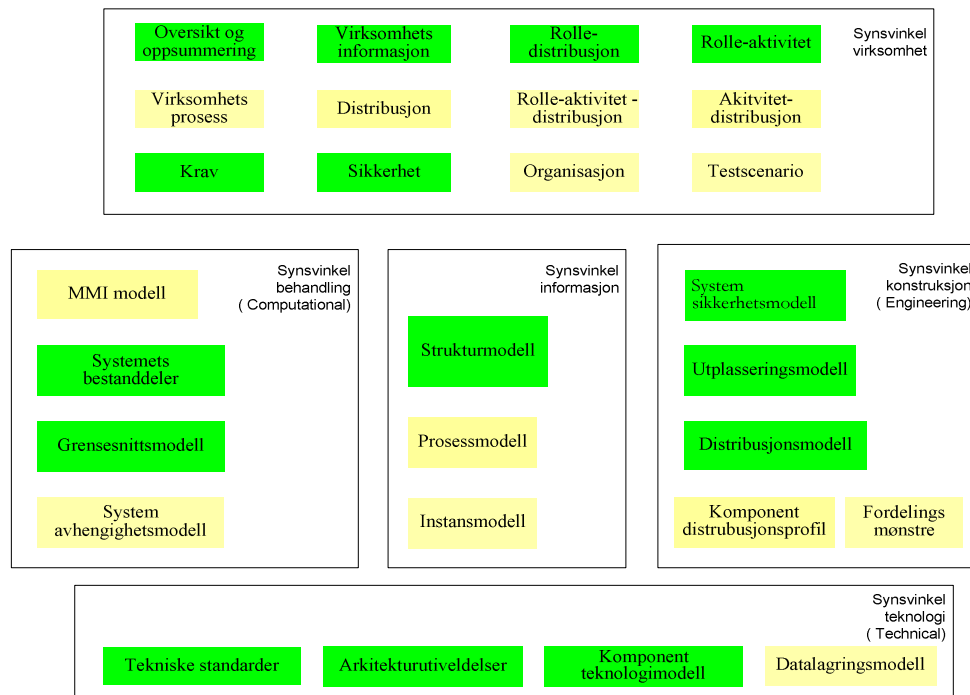
Figur 43 viser avhengighetene mellom modellene på de forskjellige nivå. Dette viser at kontekstmodellen til målsystemet kan ha avhengighet til alle modellene til virksomhetsarkitekturen, dog virksomhetens kontekst unntatt. Videre viser det at alle deler av målsystemet, kontekstmodellen unntatt, kan referere og bygge på arkitekturmodellene til subsystemet.



Figur 43 Avhengigheter mellom arkitekturmodellene

5.2 MACCIS ARKITEKTURMODELLER

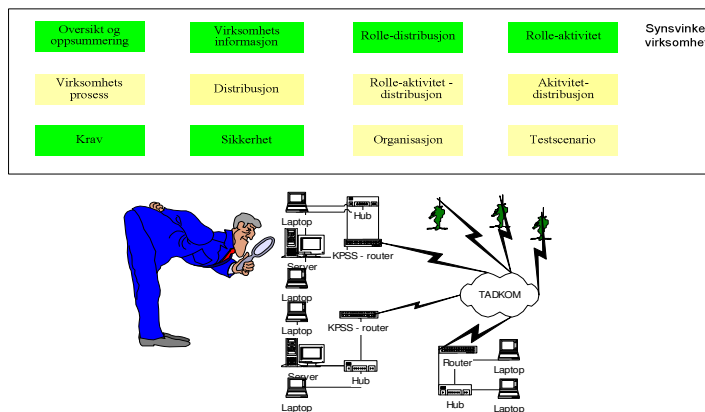
Her beskrives MACCIS arkitekturmodellene, gjennomgangen er gjort i fra de fem synsvinklene.



Figur 44 MACCIS arkitekturmodeller

5.2.1 SYNSVINKEL VIRKSOMHET

Synsvinkel virksomhet skal beskrive de nødvendige aspektene ved virksomheten kommando- og kontrollsystem (K2S) for å lage et kommando- og kontroll informasjons system (K2IS) som understøtter virksomheten. For å gjøre dette er det i MACCIS listet i alt 12 modeller. Dette er EV1 til EV12 se Tabell 4 Virksomhetsvinkelen. Dette skal være et syn på systemet sett fra virksomheten.



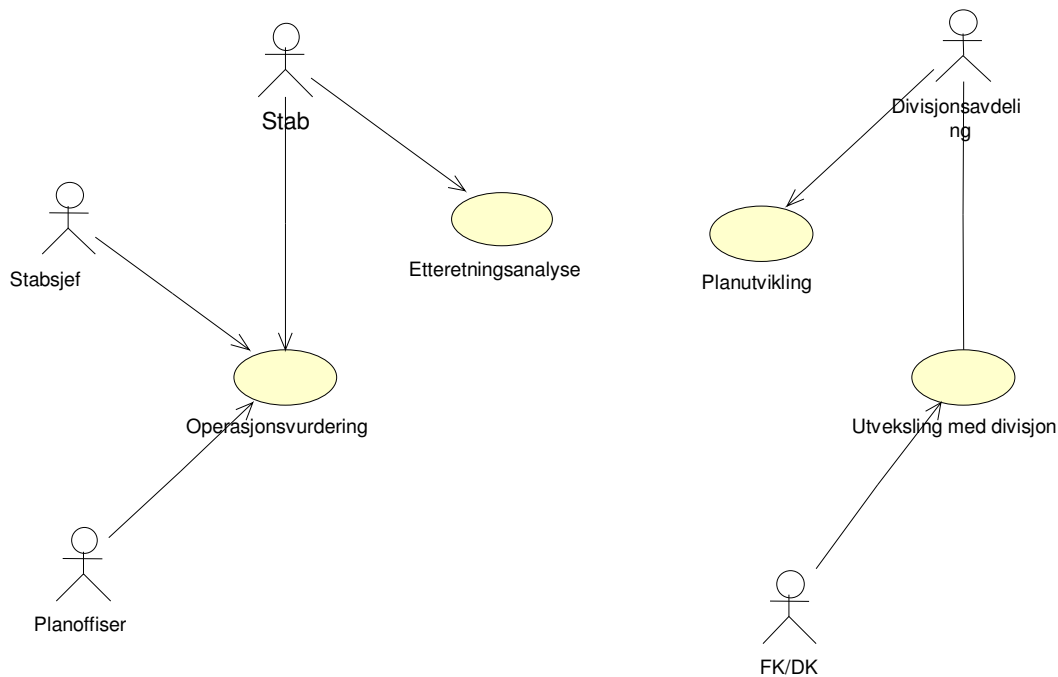
Figur 45 Synsvinkel virksomhet

Synsvinkel	Modell referanse	Modell navn	Vesentlig eller tillegg	Beskrivelse
Virksomhet	EV1	Oversikt og oppsummering	Vesentlig	Hensikt, skoping, målgruppe, omgivelser, overordnet grafisk fremvisning av operativt konsept
Virksomhet	EV2	Virksomhetsprosess	Tillegg	Aktiviteter, sammenheng mellom aktiviteter, rammer og betingelser for aktivitetene, informasjonsflyt mellom aktiviteter.
Virksomhet	EV3	Virksomhetsinformasjon	Vesentlig	Virksomhetens informasjonsmodell, - Virksomhetens informasjonsobjekter og deres relasjoner.
Virksomhet	EV4	Organisasjon	Tillegg	Organisatoriske roller, deres ansvar, deres samhandling og relaterte regler.
Virksomhet	EV5	Distribusjon	Tillegg	Noder i virksomheten
Virksomhet	EV6	Rolle-aktivitet	Vesentlig	Roller i organisasjonen med ansvar, uttrykt ved aktiviteter som skal utføres.
Virksomhet	EV7	Rolle-aktivitet - distribusjon	Tillegg	Organisatoriske roller med aktiviteter knyttet til organisasjonens noder.
Virksomhet	EV8	Sikkerhet	Vesentlig	Høyeste ordens sikkerhetspolicy
Virksomhet	EV9	Krav	Tillegg	Systemkrav
Virksomhet	EV10	Testmodell	Tillegg	Testmodell for systemkrav, inkludert brukstilfeller
Virksomhet	EV11	Aktivitet-distribusjon	Tillegg	Aktiviteter koblet til noder
Virksomhet	EV12	Rolle-distribusjon	Vesentlig	Roller koblet til noder

Tabell 4 Virksomhetsvinkelen

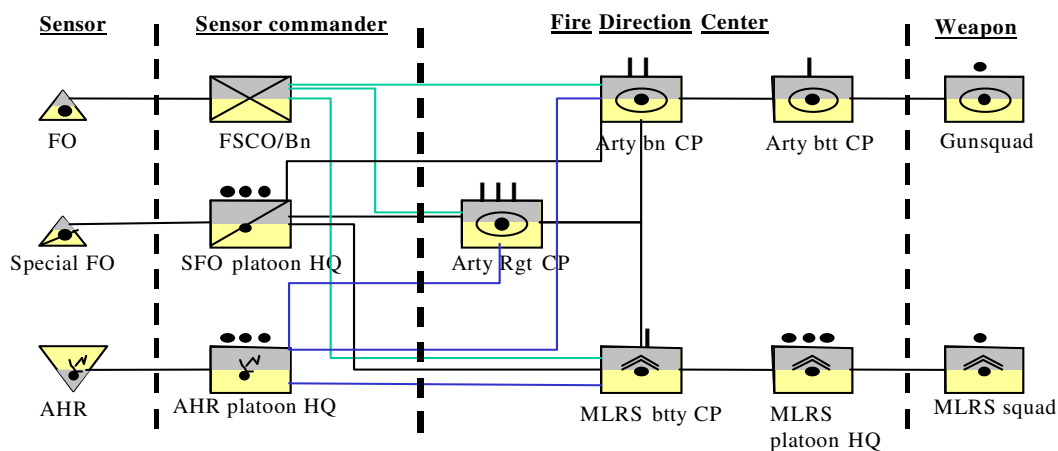
5.2.1.1 EV1 Oversikt og oppsummering

EV1 – fokuserer på avgrensning og hensikten til systemet, de tenkte brukerne og det operative konseptet. Et rikt bilde med uformell tekst anbefales og UML kan benyttes for å lage eksempelvis et overordnet brukstilfelle, se Figur 46 EV1 Overordnet brukstilfelle.



Figur 46 EV1 Overordnet brukstilfelle

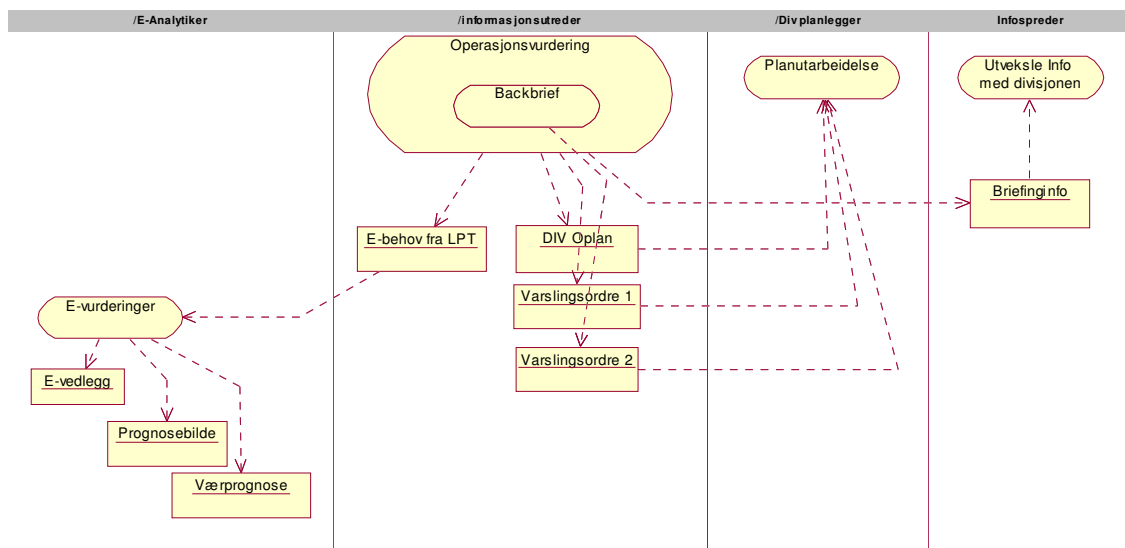
I tillegg til dette anbefaler [MACCIS] at man i EV1 legger inn uformelle tekstlige beskrivelser. K2 har mange rike grafiske symboler som er standardisert mellom alle NATOs nasjoner, disse er samlet i [APP6A]. EV1 kan derfor med fordel utfylles med bruk av symboler fra APP6a. Et eksempel på slik bruk viser samhandling og kommunikasjon i artilleriet, se Figur 47 Roller og kommunikasjon vist med APP6a symbolikk.



Figur 47 Roller og kommunikasjon vist med APP6a symbolikk

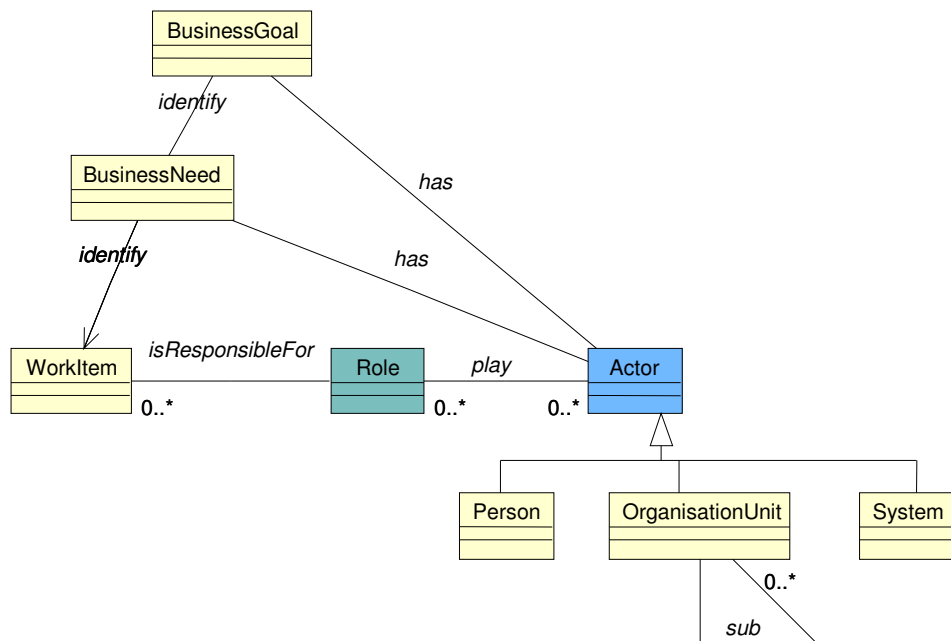
5.2.1.2 EV6 Rolle-aktivitetsmodell

EV6 – fokuserer på virksomhetsprosessen, aktivitetene som foregår i prosessen, rollene som har ansvaret for å utføre aktivitetene og informasjonen som er input eller output av aktivitetene. Målet er å forstå og formalisere aktivitetene som rollene utfører i problemområdet gitt i EV-1. UML swimlanes benyttes for å lage denne modellen, se Figur 48 EV6 Rolle aktivitetsmodell. Her er rollene i hver sin "svømmebane" De gjør egne aktiviteter innenfor sin bane, og all kommunikasjon med andre roller foregår med informasjon mellom svømmebanene.



Figur 48 EV6 Rolle aktivitetsmodell

Figur 49 Rolle-aktør gir en forklaring på forskjellen mellom rolle og aktør i denne konteksten. Roller er en svært nyttig abstraksjon i forhold til en aktør. Rollen er et sett av ansvar, mens aktører spiller roller og innehar gjennom dette ansvaret til den aktuelle rollen.



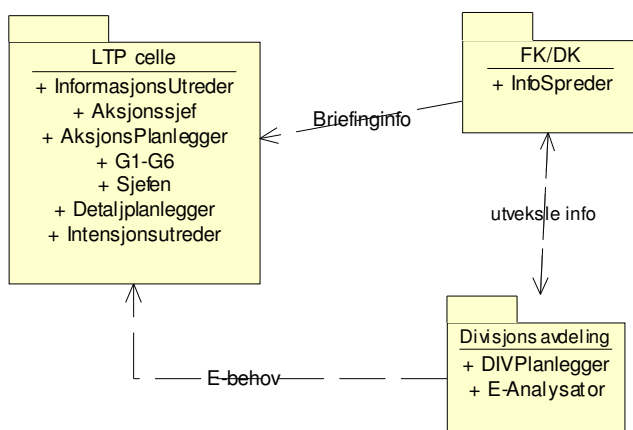
Figur 49 Rolle-aktør

Det er et mange til mange forhold mellom aktør og rolle – en aktør kan ha mange roller, og en rolle kan spilles av flere aktører. Rollen har en identitet, og er derfor unik i en gitt kontekst.

Rollene vil på et høyere nivå gjerne være sammensatt, og kan derfor brytes ned ytterligere. En slik nedbryting vil rekursivt gi en rolle-aktivitetsmodell med høyere oppløsning.

5.2.1.3 EV12 Rolle- distribusjon

Distribusjonsmodellen skal vise distribusjon av samhandlende aktører og roller. Dette illustreres med UML pakkediagram som illustrerer de distribuerte enhetene som vist i Figur 50 Rolle- distribusjon.



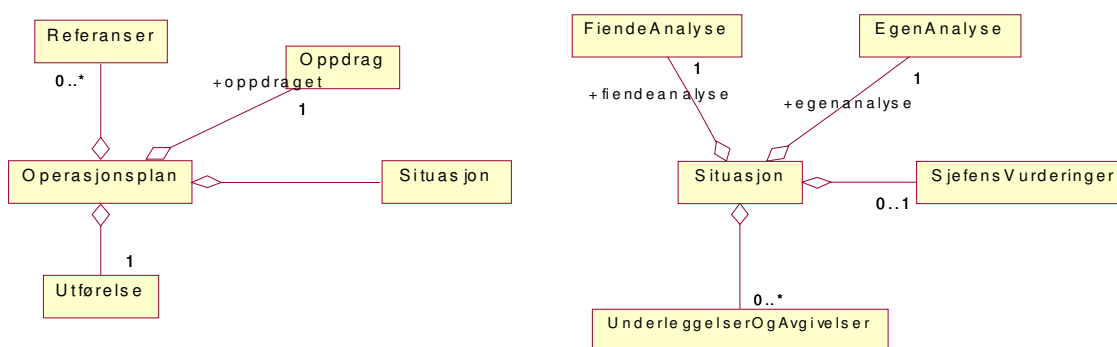
Figur 50 Rolle- distribusjon

Det kan foreligge flere behov for å illustrere dette, å synliggjøre utstrakte kommunikasjonsbehov kan være et eksempel. Gitt at man kan gjøre noe med organisasjonen vil en organisasjonsutviklingsprosess

benytte EV12 for å eventuelt strømlinjeforme enkelte prosesser. I K2S vil distribusjon være en avveining mellom sårbarhet for kommunikasjonsbrudd og distribusjon av kapasitet for å skape seighet og overlevelsessevne i organisasjonen.

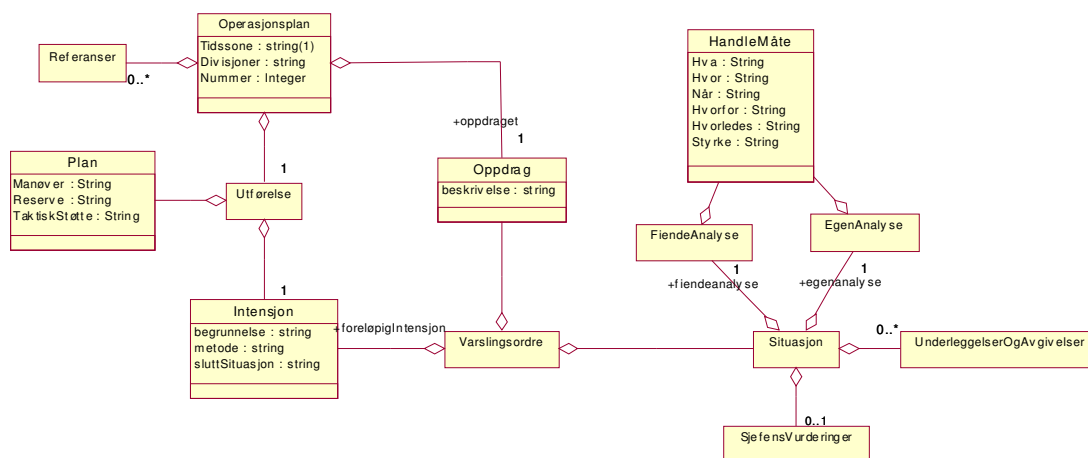
5.2.1.4 EV3 Virksomhetsinformasjon

Virksomhetsinformasjonsmodellen beskriver høynivå informasjon som er en viktig del av virksomheten. Disse elementene vil tre fram som et resultat fra EV6. Resultatene fra EV3 kan benyttes til flere ting, for eksempel kan dette være starten på informasjonsvinkelen (IV) for de informasjonselementene som skal behandles i systemet. EV3 kan også videreføres som en del av virksomheten for å lage en helhetlig informasjonsmodell for hele virksomheten. En slik prosess kan føre fram til informasjonsobjekter som har mye verdi over tid for virksomheten (ut over levetiden til det systemet vi nå har i tankene å lage) EV3 gir flere sett med UML klassediagrammer. To eksempler er vist i Figur 51 EV3 Business objekter.



Figur 51 EV3 Business objekter

I den videre utviklingsprosessen vil disse modellen tas videre til Informasjonsmodelleringen. Der vil flere detaljer tre fram. I et iterativt arbeid vil resultatet fra informasjonsmodelleringen kunne brukes for å detaljere EV3 videre. Da vil en oppdatert modell kunne bli som vist her i Figur 52 EV3 Business objekter - detaljert.



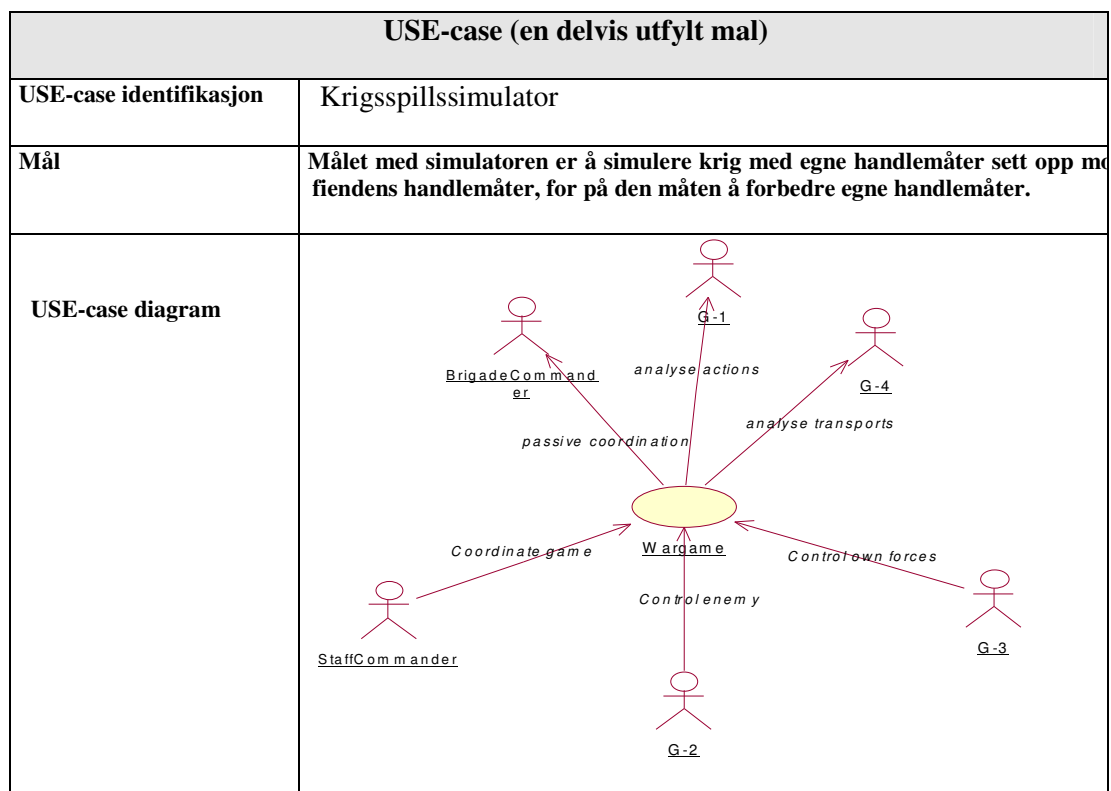
Figur 52 EV3 Business objekter - detaljert

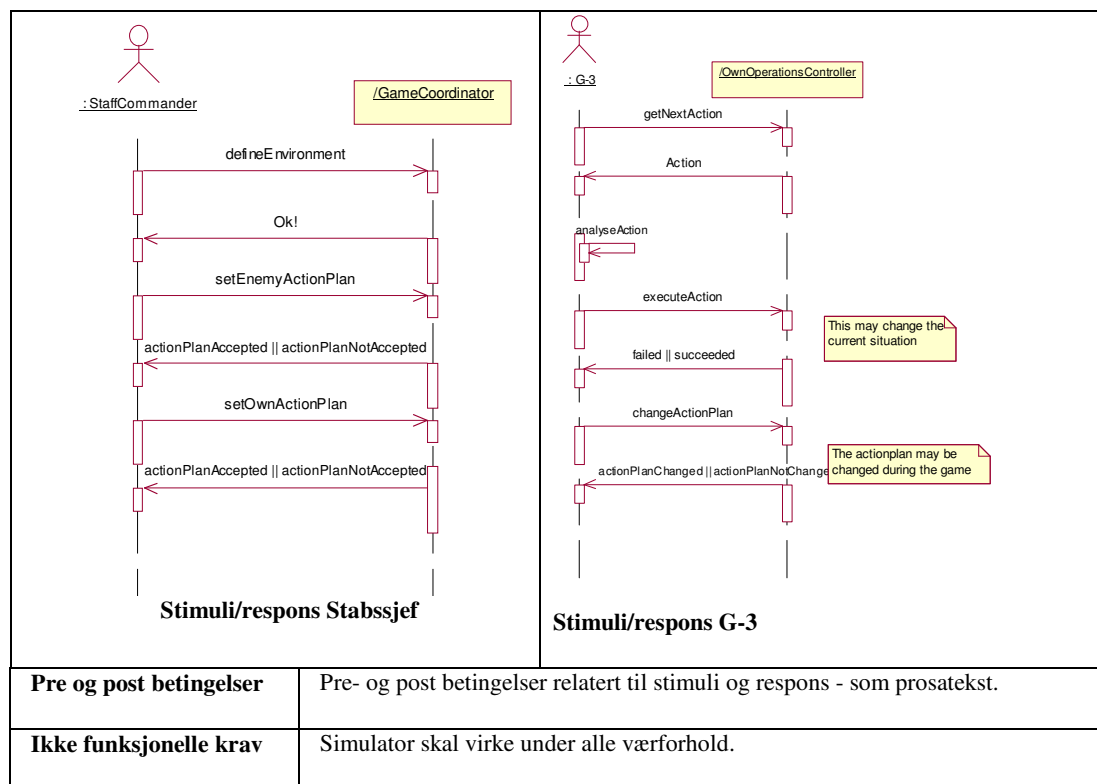
5.2.1.5 EV8 Sikkerhet

Sikkerhetsmodellen fokuserer på de høynivå sikkerhetsaspektene som systemet skal understøtte. Dette gjelder roller i systemet, det gjelder informasjon som skal flyttes og kommunikasjon mellom distribuerte enheter. Ved hjelp av modellen framkommer krav til sikkerhet for systemet, begrensninger for aktørene som skal bruke systemet og krav til hvordan informasjon skal håndteres. Modellen er i dag sterkt preget av de presentasjonskrav akkrediteringsmyndigheten Norsk Sikkerhetsmyndighet stiller. En modell vil kanskje kunne oppsummere disse formkravene, men den er ikke laget på dette stadiet.

5.2.1.6 EV9 Krav

Kravmodellen fokuserer på kravene til de informasjonssystem som skal bygges etter modellene fra virksomheten. Kravene er sånn sett bindeleddet mellom virksomheten og systemet. Skal dette fungere som en kontrakt med systemutvikleren, må dette inkludere krav hvordan aktivitetene utføres, informasjonen som skal behandles, QoS krav og sikkerhetskrav. Kravene framkommer fra UML USE-case diagrammer med tilhørende detaljer som inngangsverdier/utverdier, QoS krav. Dette er vist her som en mal i Tabell 5 Mal for use case.





Tabell 5 Mal for use case

Hvert USE-case understøttes av sekvensdiagram som her vist for den enkelte aktør.

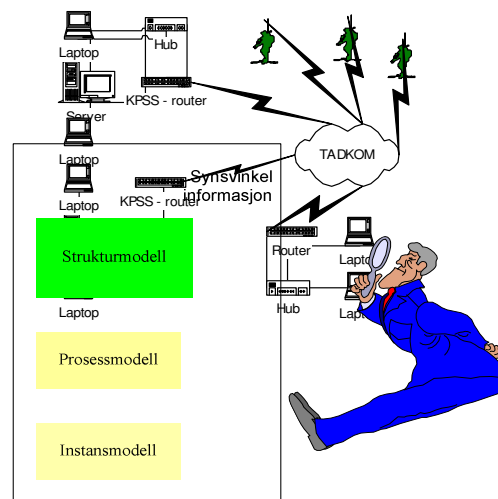
5.2.1.7 Tilleggsmodeller

De foregående seks modellene er kjernen i synsvinkel virksomhet.

I tillegg har [MACCIS] definert tilleggsmodeller. Disse modellene er ikke ment å være obligatoriske, men kan være til støtte. Særlig EV-4 Organisasjonskart antas å være nyttig. Dette er det bildet som virksomheten er vant til å bruke for å illustrere sammenhengen mellom aktørene på en hierarkisk måte.

5.2.2 SYNSVINKEL INFORMASJON

Synsvinkelen skal beskrive de nødvendige objektene og behandlingen av informasjonen i systemet. Informasjonsmodellene tar utgangspunkt i de enheter som framkommer fra virksomhetsmodellene, særlig EV3, men også aspekter fra synsvinkel system. For å gjøre dette er det i MACCIS listet 3 modeller, en vesentlig og to tilleggsmodeller, henholdsvis IV2 samt IV3 og IV1. Se Tabell 6 Synsvinkel informasjon.



Figur 53 Synsvinkel informasjon

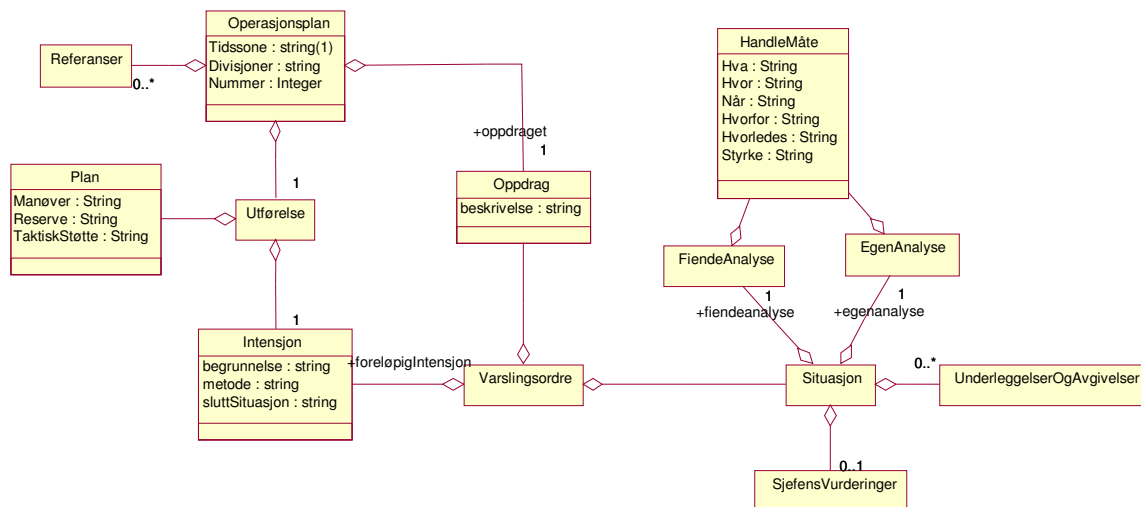
Synsvinkel	Modell referanse	Modell navn	Vesentlig eller tillegg	Beskrivelse
Information	IV1	Instansmodell	Tillegg	Regler som er gyldig på bestemt tidspunkt
Information	IV2	Strukturmodell	Vesentlig	Relasjon mellom informasjonsobjekter som alltid er sann
Information	IV3	Prosessmodell	Tillegg	Regler for endring av informasjon over tid

Tabell 6 Synsvinkel informasjon

5.2.2.1 IV2 Strukturmodell

Strukturmodellen beskriver relasjon mellom informasjonsobjekter som alltid er sann (invarianter). Modellen uttrykkes typisk som en informasjonsmodell ved hjelp av et klassediagram eller et Entity-Relationship diagram. Strukturmodellen er en logisk modell, eller konseptuell modell uavhengig av representasjon. TV-4 viser mulig mappinger til fysisk modell. Strukturmodellen er en forfining av virksomhetsinformasjon (EV-3)

Figur 54 IV2 Informasjonsstrukturmodell viser et eksempel på en strukturmodell.



Figur 54 IV2 Informasjonsstrukturmodell

5.2.2.2 Tilleggsmodeller

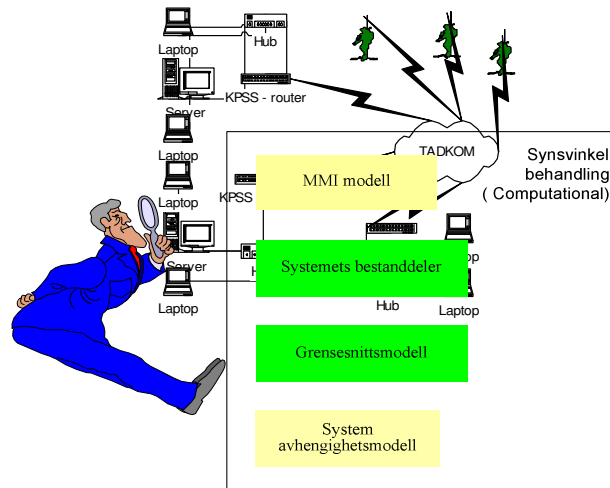
IV1 Instansmodell beskriver regler som er gyldige på bestemte tidspunkt, f. eks ved initialisering. Modellen er et klassediagram, med regler beskrevet i OCL.

IV3 Prosessmodell beskriver hvordan informasjonen utvikler seg over tid. Et eksempel kan være hvordan oppdateringer formidles i systemet, eller hvordan tilstandsandringer i systemet kan føre til endringer på informasjonsobjekter. Prosessmodellen er gjerne uttrykt i et tilstandsdiagram.

5.2.3 SYNSVINKEL BEHANDLING

Synsvinkel behandling²² spesifiserer objektene i systemet og hvordan disse samvirker.

Her benyttes entiteter – objekter og aktiviteter – prosesser som er identifisert i virksomhetsmodellen. Denne synsvinkelen består i MACCIS av 4 hovedmodeller se Tabell 7 Synsvinkel behandling, der to av modellene er betraktet som vesentlige.



Figur 55 Synsvinkel behandling

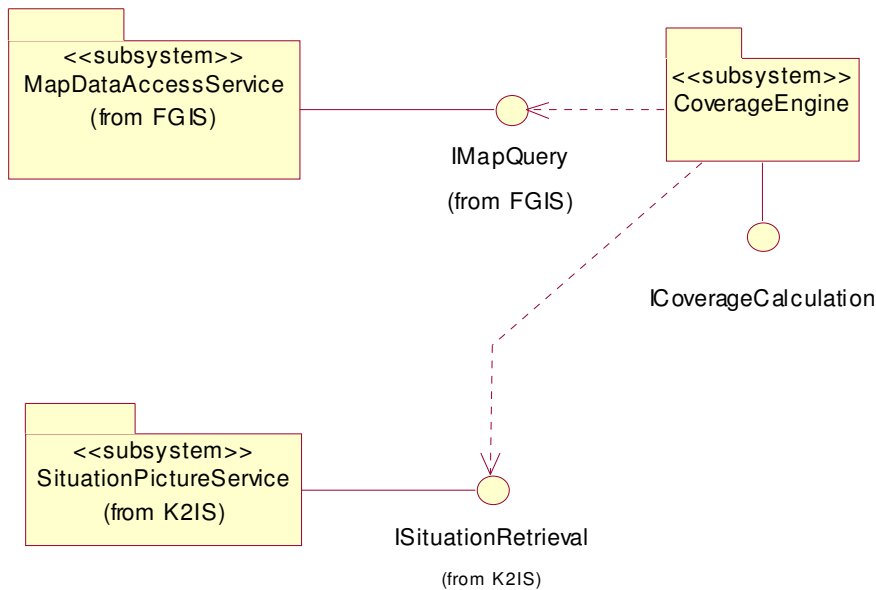
Synsvinkel	Modell referanse	Modell navn	Vesentlig eller tillegg	Beskrivelse
Behandling (Computational)	CV1	Menneske maskingrensesnitt	Tillegg	User interfaces
Behandling (Computational)	CV2	System avhengighetsmodell	Tillegg	Design models of service components
Behandling (Computational)	CV3	Grensesnittmodell	Vesentlig	En relativt detaljert beskrivelse av oppførsel og informasjon ved komponentens grensesnitt
Behandling (Computational)	CV4	Bestanddeler (System decomposition)	Vesentlig	Systemets bestanddeler collaboration

Tabell 7 Synsvinkel behandling

5.2.3.1 CV4 Bestanddeler (System decomposition)

Denne modellen beskriver systemet delt opp i sine enkelte bestanddeler, eller delsystemer, og hvordan disse delene til sammen utgjør et hele. En modularisert beskrivelse av systemet fremme gjenbruk av komponenter som er godt beskrevet. Dekomponering er et avgjørende steg i systemutviklingen, modellen er derfor vesentlig. Et eksempel på en slik modell er vist i Figur 56 CV4 Bestanddeler i systemet Felles frekvensadministrasjonssystem (FEFAS/NDFMS).

²² Behandling foreslås for Computational (var i MACCIS 1 System)



Figur 56 CV4 Bestanddeler i systemet Felles frekvensadministrasjonssystem (FEFAS/NDFMS)

Beskrivelsen skal inneholde et grensesnittssymbol for hvert definert grensesnitt.

5.2.3.2 CV3 Grensesnittsmode (Interface description model)

Denne modellen beskriver de grensesnitt som er definert til de komponenter og delsystemer identifisert i CV3. Dette skal gjøres på en slik måte at det blir entydig forstått, og slik at gjenbruk kan være mulig. En grensesnittsbeskrivelse er en relativt detaljert beskrivelse av oppførsel og informasjon ved komponentens grensesnitt. En mal for grensesnittsbeskrivelser er laget i MACCIS se [MACCIS].

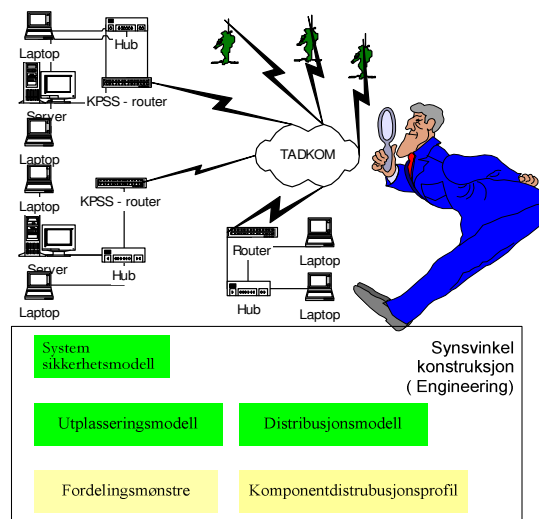
5.2.3.3 Tilleggsmodeller

Menneske maskingrensesnittsmode (CV1) er en mode for å klargjøre systemets brukegrensesnittflate. Dette er en mode for å hjelpe kommunikasjon mellom utviklere og brukere av systemet. Siden modellen bare er et kommunikasjonsmiddel, er den ikke ansett som vesentlig. Når den lages bør den benyttes for å underbygge forståelsen av kravene (EV-9) og linkes til denne for å hindre duplisering av informasjon.

Systemavhengighetsmode (CV2) skal beskrive hvordan systemet vi nå beskriver passer inn i et større hele av andre systemer. Modellen spesifiserer ytterligere de delsystemer identifisert under Bestanddeler (CV1) og som er eksterne til systemet vi nå beskriver. Modellen er derfor i seg selv ansett som å være ett tillegg.

5.2.4 SYNSVINKEL KONSTRUKSJON

Synsvinkel konstruksjon²³ tar for seg to hovedaspekter i MACCIS. Dette er sikkerhetsmekanismer i systemet og distribusjon til noder. Synsvinkelen består i MACCIS av totalt 5 modeller, der tre anses som vesentlige, se Tabell 8 Synsvinkel konstruksjon.



Figur 57 Synsvinkel konstruksjon

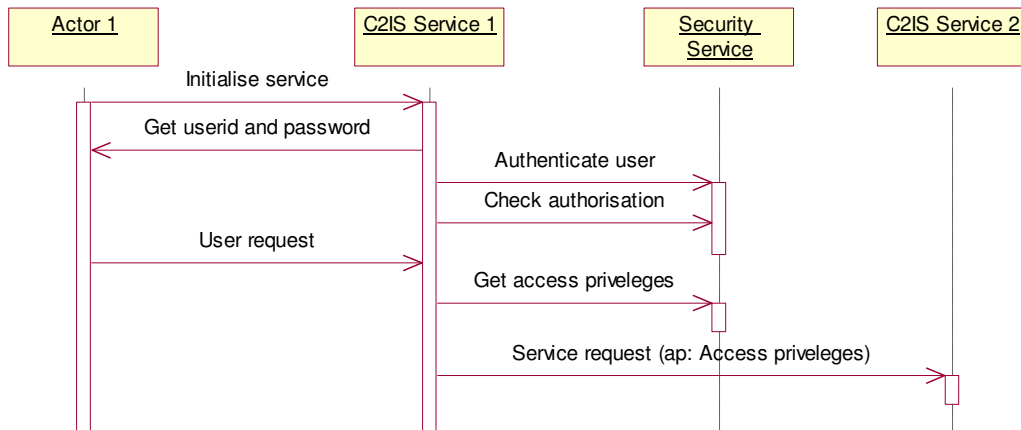
Synsvinkel	Modell referanse	Modell navn	Vesentlig eller tillegg	Beskrivelse
Konstruksjon	En-1	System sikkerhetsmodell	Vesentlig	Sikkerhetsmekanismer som understøtter sikkerhetskravene til systemet
Konstruksjon	En-2	Fordelingsmønstre	Tillegg	Generelle løsninger for distribusjon
Konstruksjon	En-3	Utplasseringsmodell	Vesentlig	Utplassering av software på fysiske noder
Konstruksjon	En-4	Distribusjonsmodell	Vesentlig	Komponenter distribuert på logiske noder.
Konstruksjon	En-5	Komponent distribusjonsprofil	Tillegg	Designkrav for komponenter i systemet på en teknologi uavhengig måte

Tabell 8 Synsvinkel konstruksjon

5.2.4.1 System sikkerhetsmodell (En-1)

Hensikten med modellen er å beskrive de forskjellige sikkerhetsmekanismer som er benyttet i systemet. Modellen består av en beskrivelse av et sett med sikkerhetsmekanismer, og UML-diagrammer som viser hvordan disse er benyttet i systemet. Et eksempel her vist i Figur 58 En-1 Autorisasjon og tilgangskontroll.

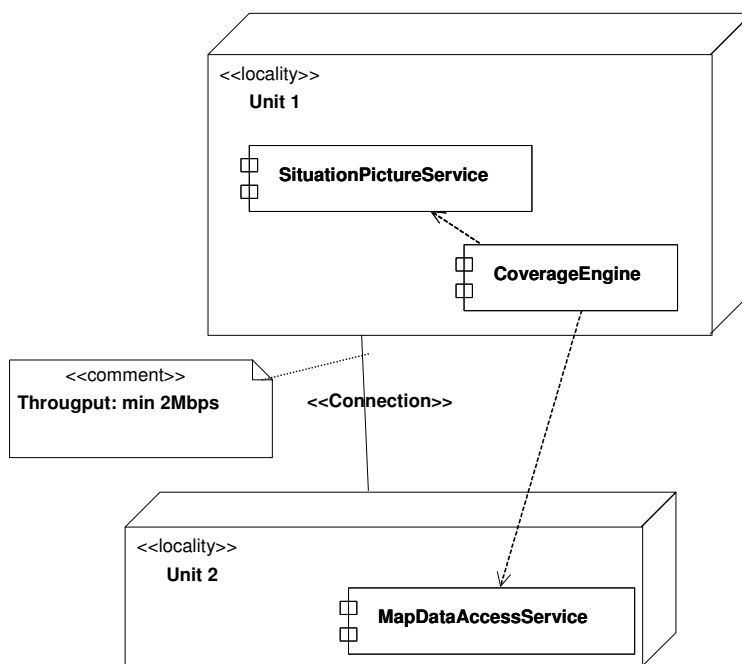
²³ Foreslår å benytte ordet konstruksjonsvinkel for (Engineering Viewpoint)



Figur 58 En-1 Autorisasjon og tilgangskontroll

5.2.4.2 Distribusjonsmodell (En-2)

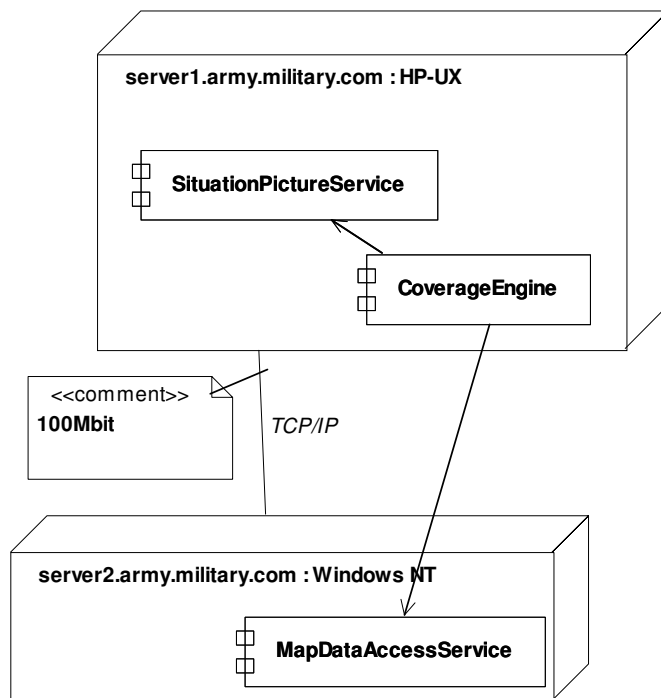
Hensikten med modellen er å beskrive logiske sett og delsystemer som må være distribuert og anvendt sammen. Modellen består av et UML deploymentdiagram som viser den logiske distribusjon. Samlingen av enheter kalles "noder" og relasjonen mellom disse kalles "connection". Et eksempel på en slik modell er vist i Figur 59 En-2 Distribusjonsmodell.



Figur 59 En-2 Distribusjonsmodell

5.2.4.3 Utplaseringsmodell (En-4)

Hensikten med modellen er å beskrive fysiske sammenhenger mellom software og hardware i systemet. I tillegg til å faktisk beskrive den eksisterende hardware og kommunikasjonslinker, vil denne modellen også kunne identifisere nye krav til hardware og kommunikasjonsløsninger. Et oppfølgingseksempel fra forrige modell er vist i Figur 60 En-4 Utplaseringsmodell fra FEFAS/NDFMS



Figur 60 En-4 Utplaseringsmodell fra FEFAS/NDFMS

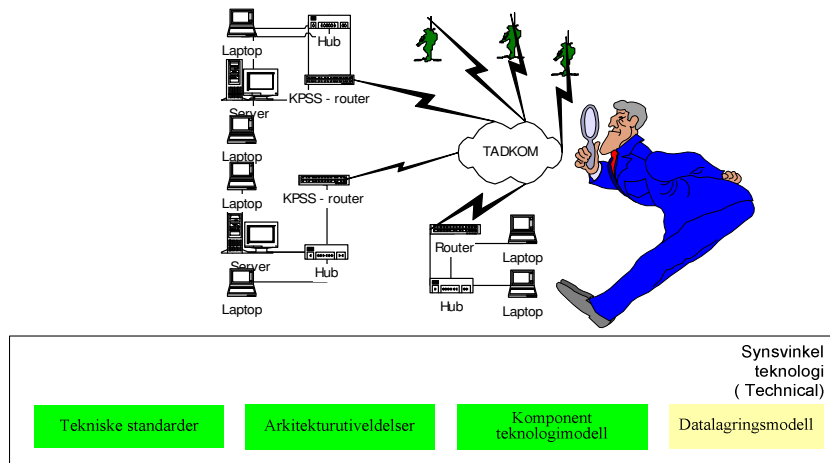
5.2.4.4 Tilleggsmodeller

Fordelingsmønstre En-2 (Distribution patterns) er generelle løsninger for fordeling basert på ODP-definerte løsninger for transparens. Modellen beskrives med UML-collaboration diagrammer.

Komponentdistribusjonsprofil (En-5) beskriver designkrav for komponenter i systemet på en teknologi uavhengig måte. Denne modellen danner naturlig basis for det teknologispesifikke design i synsvinkel teknologi.

5.2.5 SYNSVINKEL TEKNOLOGI

Teknologivinkelen tar for seg den benyttede teknologi, dette innebærer et sett av aspekter. Selve systemet er en del av denne synsvinkelen, representert ved kjørbare kode og infrastruktur benyttet, se for øvrig Tabell 9 Teknologivinkelen.



Figur 61 Teknologivinkelen

Synsvinkel	Modell referanse	Modell navn	Vesentlig eller tillegg	Beskrivelse
Teknisk	TV1	Tekniske standarder	Vesentlig	Liste av brukte av standarder
Technical	TV2	Arkitekturutvidelses modell	Vesentlig	Utvidelser til arkituren (CV-2, CV-4) for valgt teknologi
Technical	TV3	Datalagringsmodell	Tillegg	Logisk datamodell for valgt lagringsløsning
Technical	TV4	Komponentteknolog iprofil	Vesentlig	Detaljert design og implementering i valgt teknologi.

Tabell 9 Teknologivinkelen

5.2.5.1 Tekniske standarder (TV-1)

Tekniske standarder er forhåpentligvis lagt til grunn ved utviklingen av systemet. Her listes disse på en entydig måte. Et eksempel på en slik liste er vist i her.

1. ISO/IEC 2382-1: 1993, *Information technology - Vocabulary - Part 1: Fundamental terms*
2. ISO 8601: 1998, *Data elements and interchange formats - Information interchange - Representation of dates and times*
3. XML Schema Part 1: Structures, W3C Recommendation 2 May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502>

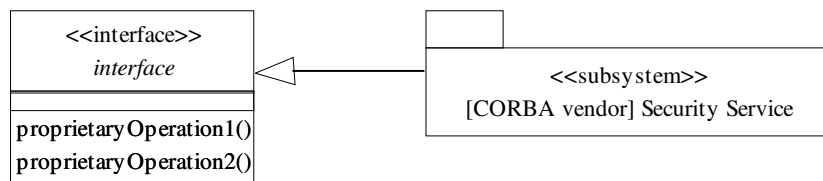
Figur 62 TV-1 Tekniske standarder

5.2.5.2 Arkitekturutvidelsesmodell (TV-2)

Sikkerhets og distribusjonsegenskaper som tidligere er adressert i synsvinkel konstruksjon skal vise de standardmetoder som er benyttet. Ideelt sett skal derfor implementeringen av disse tjenestene følge de foreskrevne standarder. Det er likevel sjelden at valgt teknologi følger standarder, og det kan være eksplisitte proprietære grensesnitt som er benyttet.

Denne modellen beskriver derfor dette, og i de tilfeller der standarden er benyttet, uttrykkes dette eksplisitt – det utelates ikke å kommentere dette.

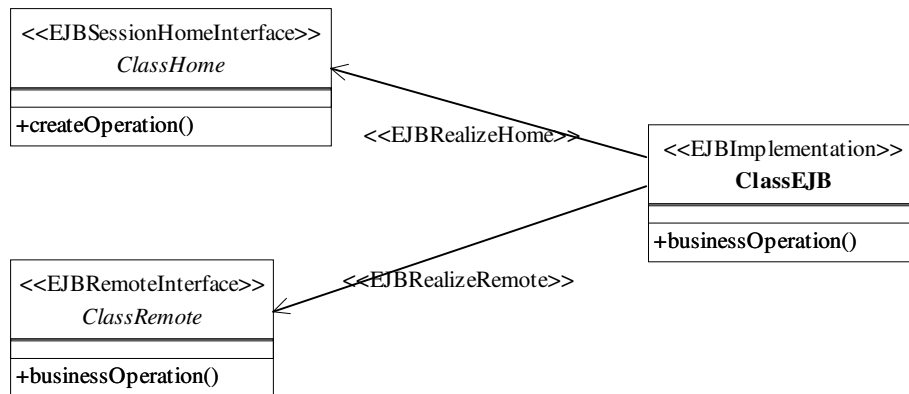
Et eksempel på en utvidelse er vist i Figur 63 TV-2 Arkitekturutvidelsesmodell, som beskriver et proprietært grensesnitt til en spesifikk implementering av en sikkerhetstjeneste.



Figur 63 TV-2 Arkitekturutvidelsesmodell

5.2.5.3 Komponentteknologiprofil (TV-4)

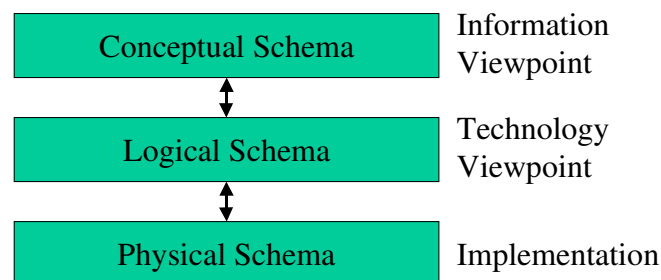
Komponentteknologiprofil erstatter det som tidligere var ”kode” i en implementeringsmodell. Hensikten med modellen er å gi en detaljert oversikt over design og implementering i valgt teknologi. For hver komponent skal den eksterne UML-profil for valgt teknologi beskrives. Et eksempel på dette er vist i Figur 64 TV-4 Komponentteknologiprofil - et eksempel for EJB.



Figur 64 TV-4 Komponentteknologiprofil - et eksempel for EJB

5.2.5.4 Tilleggsmodell

En datamodell kan dele i tre forskjellige skjema, som vist i Figur 65. Det konseptuelle skjema viser informasjonen slik den ser ut fra systemet, og modelleres i synsvinkel informasjon. Det logiske skjema representerer designskjema for implementering, og mapper dermed det konseptuelle skjema til et logisk skjema i databasesystem. Det fysiske skjema representerer den fysisk lagrede representasjonen i en database.



Figur 65 Konseptuelt, logisk og fysisk skjema

Datalagringsmodell (TV-3) har til hensikt å beskrive datamodellen på et nivå hvor den kan vedlikeholdes over tid. Det fysiske skjema og mapping til det logiske skjema bør håndteres i et valgt datasystem fra en leverandør. Modellen er dermed et logisk skjema, som vil variere med valgt implementering for informasjonslaget.

6 BRUK AV MODELLER I SYSTEMUTVIKLING

I begrepet systemutvikling ligger ordet system og ordet utvikling. Ordet system defineres som ^[AschGyld]

- ”Ordnert sammenstilling av deler til et hele; planmessig fremgangsmåte. ”eller
- ”Sett eller gruppe av gjenstander eller av deler som hører sammen eller er ment å virke sammen etter en bestemt plan”

Begge definisjonene tillegger ordet system et sett av elementer samt en bestemt orden og struktur. Utvikling av et system kan da være å endre på gruppen av gjenstander og - eller endre på orden og struktur. For å holde en oversikt er det vanlig å lage seg en eller annen modell.

I definisjon 1 ligger det en planmessig fremgangsmåte, dette er også sentralt i deltermen ”utvikling”. Hvordan gjøres en utvikling planmessig? Jeg har i kapittel 4.1.4 sett litt på hva jeg kjenner til av utviklingsmetoder. Det ligger noen krav til slike metoder i kapittel 3.2. Hvordan forholder MACCIS seg til disse kravene?

Jeg beskriver her noen trekk ved utviklingen av MACCIS, deretter beskriver bruk av arkitekturmodeller for systemmodellering og hvordan slike arkitekturmodeller kan håndteres over tid. Til slutt presenteres hvordan MACCIS kan brukes for å håndtere anskaffelser. Denne presentasjonen er stort sett bygget på [MACCIS].

6.1 UTVIKLING AV MACCIS

MACCIS ble i første versjon utviklet som en spesialisering av RM-ODP med inspirasjon å ta over for C4ISR-AF, som til da var benyttet i studiearbeide i utvikling av Hærens K2IS.

6.1.1 OPPSUMMERING ETTER MACCIS VERSJON 1

Standardisering og fleksibilitet går ikke alltid sammen. Standarder er ofte oppfattet som tvangstrøyer. Derimot vil behovet for samvirke gjøre bruken av standarder nødvendig. Slik er det med Informasjonsinfrastrukturen K2IS. Internasjonalt standardiseringsarbeid er ofte sene prosesser. Det er mange parter som skal synkroniseres der de er i ulike faser av å utvikle sine systemer. Endringer vil alltid måtte påregnes. Spørsmålet som er blitt stilt er da om standardisering hindrer fleksibilitet. Og hvordan kan man eventuelt hindre dette. Et forslag er bruk av rammeverk. En slik modell for beskrivelse av systemet er ment å ivareta kravene til modularisering samtidig som man kan og skal følge standarder. Modularisering er sett som den viktigste metoden for å sikre fleksibilitet. På denne måten kan bruken av rammeverk innenfor K2IS være et middel til å oppnå bruk av standarder samtidig som man kan ha fleksibilitet.

6.1.2 VIDEREUTVIKLING AV PROSESS

For å være en rettesnor – kanskje særlig i tidlige faser av innføring bør MACCIS følges av med en prosess. En av de største svakhetene i C4ISR var nettopp hvordan modellens skulle utvikles, deres rekkefølge og prioritet. MACCIS 1 introdusert Rational Unified Process, som i dag er den mest fremherskende software utviklingsprosessen. MACCIS 1 sa relativt lite om hvordan modellering etter MACCIS skulle forholde seg til den iterative prosessen i UP. Videre var det svært mange modeller som var ”mandatory” – slik at det virket relativt omfattende å ta rammeverket i bruk. Hvis tanken blir ”alt eller intet” – slik at man ikke tar rammeverket i bruk dersom man ikke har store ressurser satt av

til arkitekturmodellering i prosjektet, mener jeg rammeverket kan virke mot sin hensikt. En slik tanke skyldes manglende forståelse for hva bruk av rammeverket kan gi av positive resultater. Dette er et ledelsesansvar. I den første studiefasen av Hærens K2IS ble dette drøftet relativt grundig. Det ble laget et godt pedagogisk dokument som presenterte evolusjonær utvikling.

Manglende detaljer på utviklingsprosess var blant annet bakgrunnen for at MACCIS ble videreutviklet til versjon 2. I det følgende presenteres hoveddeler av den systemutviklingsprosess som nå ligger med [MACCIS].

6.2 SYSTEMUTVIKLINGS- OG ANSKAFFELSESPROSESS

Her presenteres en utviklingsprosess bestående av 3 delelementer

- Virksomhetsmodellering
- Systemmodellering
- Modelldrevet anskaffelse

Disse prosessene har henholdsvis fokus på å innhente krav, overordnet arkitektur og prosjektledelse. Denne beskrivelsen er i stor grad basert på MACCIS II[MACCIS].

6.2.1 VIRKSOMHETSMODELLERING

6.2.1.1 Virksomhetsmodellering i Forsvaret

Domenespråket er basis for forståelse av virksomheten. Dette er det standardiserte språk som anvendes mellom domeneeksperter (brukere), domene analytikere og systemutviklere. Domenespråket er et middel for å kommunisere brukernes behov til modelleksperter, og som dermed kan anvendes av systemutviklerne. I forsvarssektoren har det vist seg å være nødvendig å formalisere samkvem mellom de utførende rollene i denne analyseprosessen. Dette vil kunne avhjelpe en i dag uklar prosess, med mange aktører og svært mange og varierende typer av leveranser

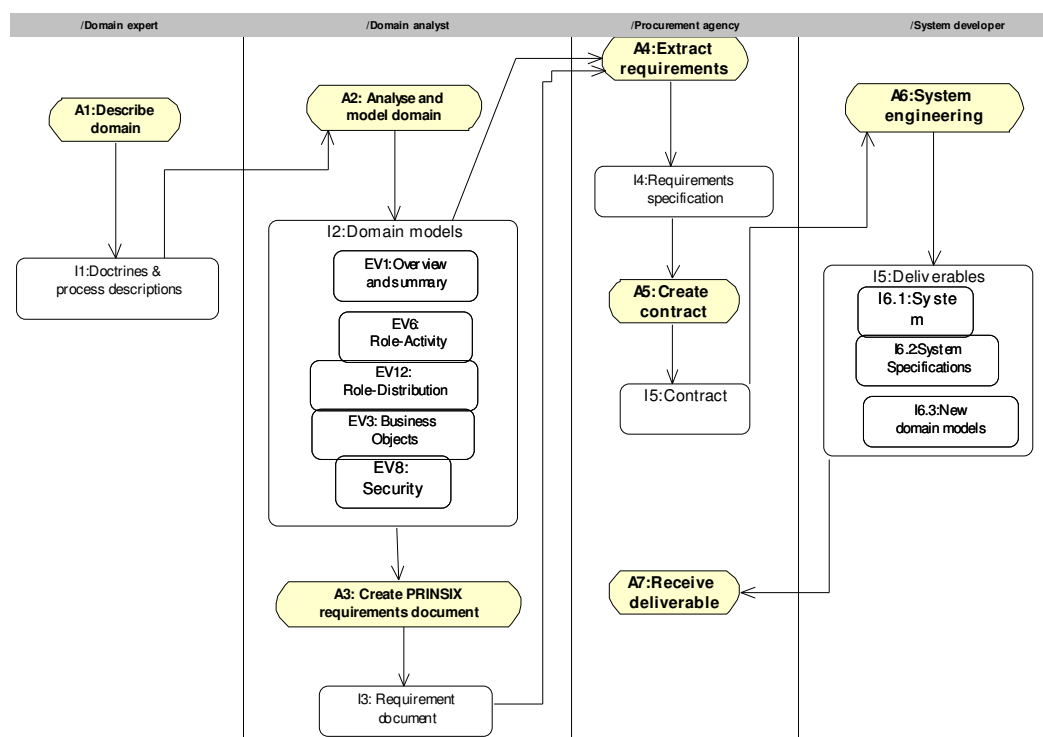
De sentrale rollene i dette spillet er i hovedsak ikke endret av de senere organisasjonsendringer i Forsvaret. Rollene er:

- Anskaffelsesorganisasjonen (Forsvarets Logistikkorganisasjon)
- Domene ekspertise (Forsvarets Stabsskole, TRADOK, LUKS, etc.)
- Domene analytikere (Forsvarets kompetansesenter for kommando og kontrollinformasjonssystemer (FK KKIS) som K2IS koordinator)
- Systemutviklere (Leverandørindustrien, f.eks. CCIS House)

Den enkeltes ansvarsområder er gjengitt her:

Anskaffelsesorganisasjon:	Utarbeide system kravspesifikasjon og kontrahere fra systemutvikler på basis av kravdokumenter
Domene eksperter:	Ivareta og fremskaffe detaljert informasjon om forsvarssektoren, hvordan de enkelte prosesser gjennomføres. Dette dokumenteres i doktriner og gjøres tilgjengelig for domene analytikere.
Domene	Beskriver formelt en domenemodell av forsvarsprosessene. Formen må

analytiker:	være av en slik karakter at innholdet er lettfattelig for aktører og i tråd med det valgte rammeverket (her MACCIS) Det skal være mulig å utlede kravspesifikasjoner basert på denne modellen.
System utvikler:	Utvikle et system som ivaretar kravene. Utarbeide en ny "as-is" beskrivelse (domenemodell) som en leveranse til anskaffelsesorganisasjonen, der nye endringer som følge av nytt system er ivare tatt.



Figur 66 Forsvarets overordnede "livssyklus"

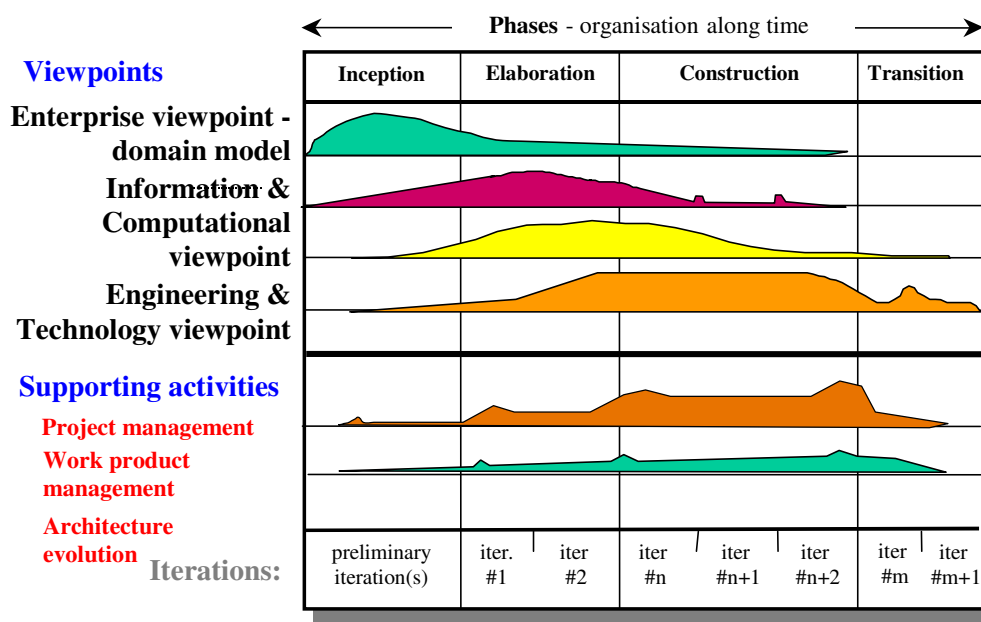
Figur 66 viser en overordnet modell av livssyklusen i Forsvaret, her med fokus på kravdomenet. Den enkelte aktivitet fører til fremtakelse av eksplisitte artefakter (MACCIS). Den beskrevne prosessen i MACCIS er basert på Unified Process. For at denne skal være anvendelig trengs en ytterligere presisering for å være anvendelig til dette formålet (virksomhetsmodellering)

I det videre følger en mer detaljert beskrivelse av analyseprosessen for å lage en domenemodell (kapittel 6.2.1.2) og utvikling av krav (kapittel 6.2.1.3)

6.2.1.2 Prosessen

Domeneanalysen inngår som en del av systemutviklingen. Unified Process som basis for denne prosessen i MACCIS er et godt utgangspunkt som rammeverk for systemutviklingen.

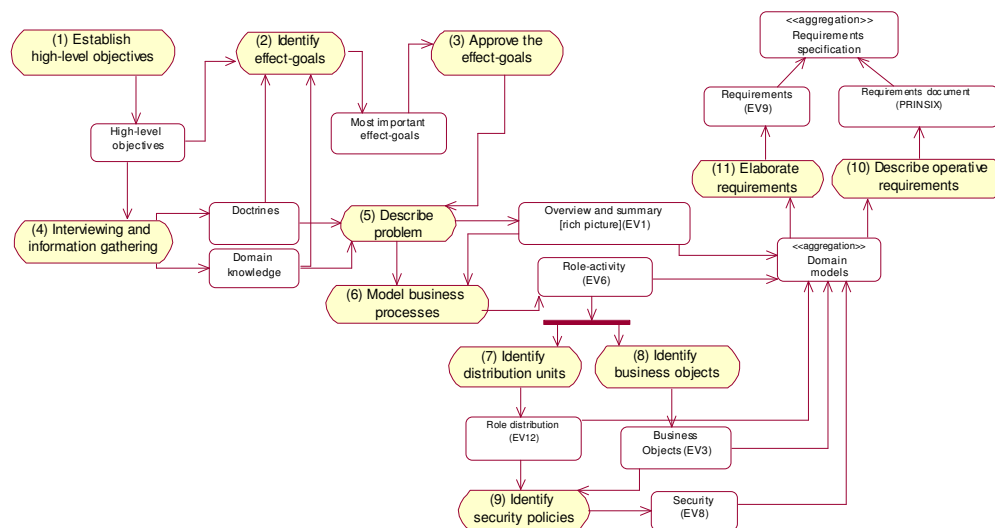
Virksomhetsmodelleringen fokuserer på de essensielle modeller som skal tas frem. Figur 67 viser den overordnede MACCIS prosessen. Her foreskrives en iterativ utviklingsprosess, delt opp i mindre elementer kalt faser. Hver fase består av et sett med iterasjoner. MACCIS synsvinkler har elementer som i varierende grad hører hjemme i alle iterasjoner.



Figur 67 Overordnet MACCIS prosess

6.2.1.2.1 Beskrivelse av prosessen

Proessen knyttet til domeneanalyse leverer et sett med essensielle modeller. Figur 68 viser et UML aktivitetsdiagram med aktiviteter og leveranser i denne prosessen.. Numrene viser til elementer som beskrives ytterligere i det følgende.



Figur 68 Domeneanalyse

Fundamentet i domeneanalysen er et sett med overordnede mål (high-level objectives (or goals) (1)). Disse beskriver visjon og gjennomgående mål for virksomheten. Av dette utledes en serie med effektmål (effect-goals (2)), som er gjenstand for godkjenning (3). For å oppnå disse effektmål, må de enkelte virksomhetsprosesser analyseres og kanskje endres.

Forståelse av virksomheten oppnås gjennom intervjuer med domeneeksperter og informasjonsinnhenting (4). Denne forståelsen anvendes for å beskrive problemet (5), som igjen fører til en modell av et oversiktlig rikt bilde (MACCIS EV1). Denne modellen er grunnlaget for virksomhetsprosessene (6), som igjen er grunnlaget for distribusjon (7) og ”business objects” (8).

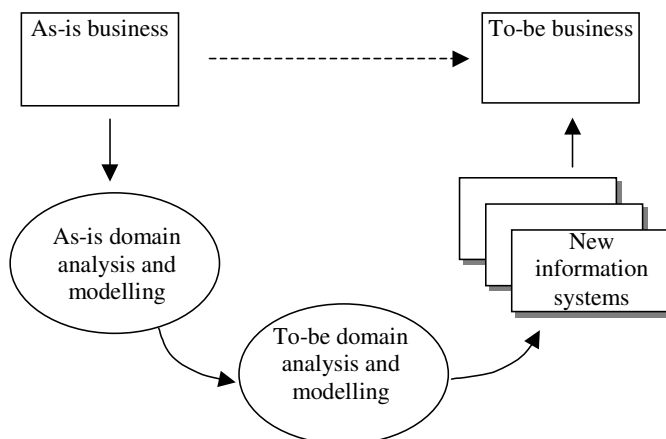
Sikkerhetsmodellen (9) tar dette som grunnlag (samt maler for Kravspesifikasjon Sikkerhet) og gir en spesifisering for sikkerhet. En helhetlig domenemodell er grunnlaget for kravmodell i MACCIS EV9 (10) og krav som f. eks kan være iht. PRINSIX (11), prosjektstyringsmodellen i Forsvaret

6.2.1.2.2 Fra As-is til To-be

Virksomhetsmodellering er todelt, as-is analyse og to-be analyse. Begge elementer kan være like essensielle å kjenne til.

As-is modellen beskriver verden som den er med prosesser og informasjonsflyt. To-be modellen beskriver verden som den er tenkt å være, ved innføring av et nytt informasjonssystem.

Innføringen av et nytt informasjonssystem vil sannsynligvis endre måten virksomheten fungerer på. Figur 69 viser hvordan virksomheten endres fra as-is til to-be ved innføring av et nytt informasjonssystem.



Figur 69 As-is vs. to-be

Domeneanalytikeren bør kjenne as-is situasjonen før fokus rettes på to-be. As-is analyse kan være en ad-hoc aktivitet eller gjøres mer formelt i form av en modell. To-be analysen er grunnlaget for å forstå virksomheten der systemet skal anvendes. To-be analysen bør beskrive hvordan virksomheten endres på bakgrunn av nytt informasjonssystem. Dette vil ikke være en eksakt vitenskap, men bør baseres på de beste mulige antagelser.

Domeneanalytikeren bør beskrive de beste mulige antagelser om virksomheten, basert på nye operative prosedyrer, ny teknologi og trender.

Målsettingen med denne aktiviteten er åpenbart å vurdere effektiviteten til en ny organisasjon. Domenemodellen kan dermed også være grunnlaget for ytterligere analyser for å optimalisere virksomheten, og igjen være grunnlaget for neste generasjon av informasjonssystemer.

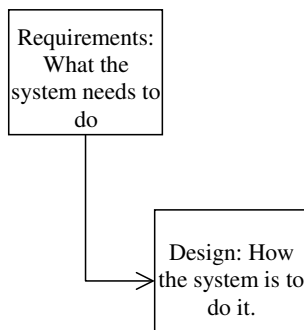
6.2.1.3 Kravhåndtering

Krav skal angi kapasiteter som systemet skal gi, og oppnåelse av disse krav er ofte et mål for om systemutviklingsprosjektet har lyktes eller ikke.

Det kan derfor være fornuftig å ha en etablert måte å utlede, organisere og håndtere krav etter hvert som disse endres.

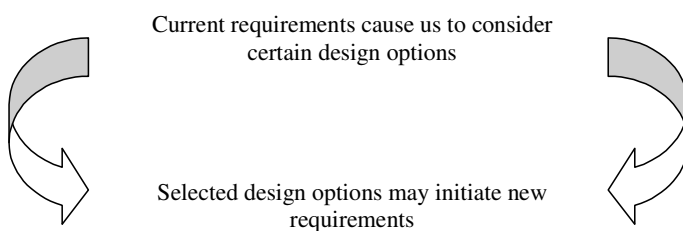
Kravhåndtering kan beskrives som en systematisk tilnærming til å utlede, organisere og dokumentere krav til et system ^[LEFFING], og en prosess som opprettholder avtalen mellom kunden og prosjektet om systemkravene ettersom de endres.

Figur 70 viser skillet mellom kravhåndtering og design. Krav beskriver hva systemet er ment å gjøre, mens design beskriver hvordan dette skal gjøres.



Figur 70 Kravhåndtering vs. Design

Utleddning av krav og dertil hørende design må gjøres i en iterativ prosess, der utledning av krav, definisjoner og designbeslutninger henger sammen. Dette passer bra inn i en iterativ utviklingsprosess og betyr at krav også kan endres og nye krav kan dukke opp under prosessens gang.



Figur 71 Rundgangen mellom krav og design

I Forsvarets utviklingsprosess kan denne rundgangen bare gjennomføres med en sterk tilbakekobling mellom systemanalytikere, brukergrupper og systemutviklere. En utfordring har her vært krav som er utviklet av K2IS koordinator og brakt videre til systemutviklere via anskaffelsesorganisasjonen. For at tilbakekoblingen skal være sterk må det være en kontinuerlig samhandling mellom systemutvikler, brukergrupper og domeneanalytiker, slik at feedback på krav kan bringes tilbake til domenemodellen.

Krav er tidligere i denne oppgaven karakterisert som funksjonelle og ikke-funksjonelle. Ikke-funksjonelle krav kalles også quality of service (QoS) (eller bare kvalitets) krav. Vi har tidligere sett at

det er en sterk sammenheng mellom funksjonelle og ikke-funksjonelle krav, fordi et komplett krav både involverer et funksjonelt krav og et assosiert sett av QoS krav.

En oppsummering kan gjøre slik:

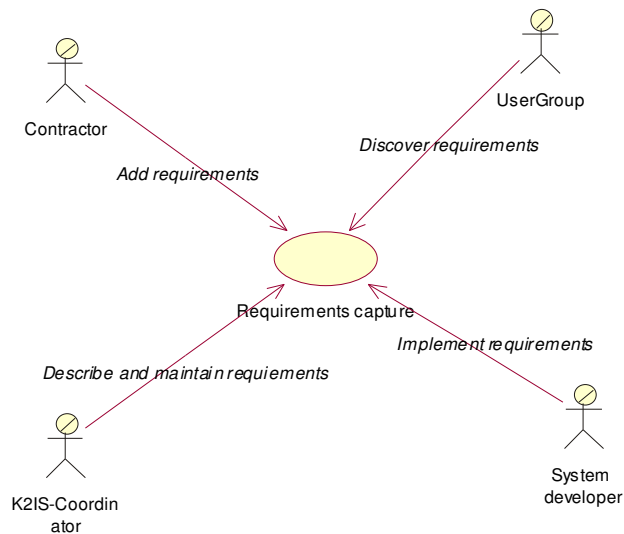
- Funksjonelle krav: Hvordan systemet oppfører seg
- Vanligvis hendelsesorientert
- USE-case
- Kvalitetskrav: Egenskaper til systemet, og dets omgivelser som ikke er funksjonelle
- Brukervennlighet (Usability)
- Pålitelighet og tilgjengelighet (Reliability, Availability)
- Feilrate (Mean time between failures (MTBF), Mean time to repair (MTTR), Maximum bugs or defect rate)
- Nøyaktighet (Accuracy)
- Ytelse (Performance, Response time for transactions, Throughput, Capacity)
- Kontrollerbar degradering (Degradation modes)
- Vedlikeholdsvennlighet (Supportability, Maintainability)
- Sikkerhet (Security)

I tillegg vil det være designkrav som setter begrensninger på systemdesign eller prosessen. Dette er som oftest etablerte standarder i virksomheten, for eksempel: Operativsystem, Komptabilitetskrav til andre systemer, grensesnitt, Programstandarder eller Etablert praksis og andre standarder i virksomheten.

For å sikre gjennomgående kvalitet til krav, angis et sett med kvalitetsmål, kraven skal være:

Korrekte, Utvetydige, Komplette, Konsistente, Rangert for betydning og stabilitet, Verifiserbare, Lette å endre, Sporbare og Forståelige.

Hvem er så ansvarlig for utesking av krav? Det er i alle tilfelle nødvendig at kravinnsamling håndteres på en konsistent og veldefinert måte. En enkelt aktør bør ha ansvaret for å vedlikeholde de operative aspekter i Forsvarssektoren. På denne måten kan det være én konsistent basis for utvikling av systemer. Kravinnhenting er prosessen for å avdekke, forstå og dokumentere krav knyttet til et gitt domene. Figur 72 viser involverte aktører og ansvarsområder i denne prosessen.



Figur 72 Kravinnhenting

Tradisjonelt har kravinnhenting og uforming vært et ansvar knyttet til våpenskoler og inspektørrollen i Forsvarets grener. For K2IS domenet har dette i Hæren vært K2IS koordinatorens rolle. Med etableringen av et felles kompetansesenter for K2IS er det der rollen som domeneanalytiker skal ligge.

I utviklingsprosessen leveres kravene fra domeneanalytikeren til anskaffelsesorganisasjonen som et kravdokument. Anskaffelsesorganisasjonen videreutvikler dette til en kravspesifikasjon for systemutvikleren. Med visuell modellering anbefales det at de tekstlige kravene så vel som domenemodellen inngår i kravspesifikasjonen. I kravanalysen vil systemutvikleren agere tett med brukergrupper for å etablere ytterligere innsikt i det operative domenet. Denne prosessen må gjennomføres i tett kontakt med domeneanalytikeren, for å vedlikeholde en oppdatert domenemodell.

SINTEF har her tillatt seg å komme med en relativt klar tilleggskommentar, som jeg mener er berettiget: "Det er verdt å merke seg at et vesentlig aspekt av kravhåndtering er faren med å gå for detaljert til verks, og derigjennom unødige holde igjen tempoet i den iterative og inkrementelle prosessen. Det er lov å prioritere og å ta rå beslutninger!"

Dette siste sier noe om prosjektledelse i denne utviklingsprosessen. Jeg vil komme tilbake til dette i neste kapittel, der jeg skisserer elementer i det jeg kaller beslutningsprosessen.

6.2.2 SYSTEMMODELLERING

6.2.2.1 Hensikt

Hensikten med systemutviklingsprosessen er å initiere, koordinere og kontrollere software utviklingen av et komponentbasert system eller individuelle komponenter som kan brukes i et slikt system. Utviklingsprosessen utføres vanligvis i et utviklingsprosjekt. Ettersom hvert enkelt prosjekt er unikt, skisseres her bare en generisk prosess som må tilpasses det enkelte prosjekt.

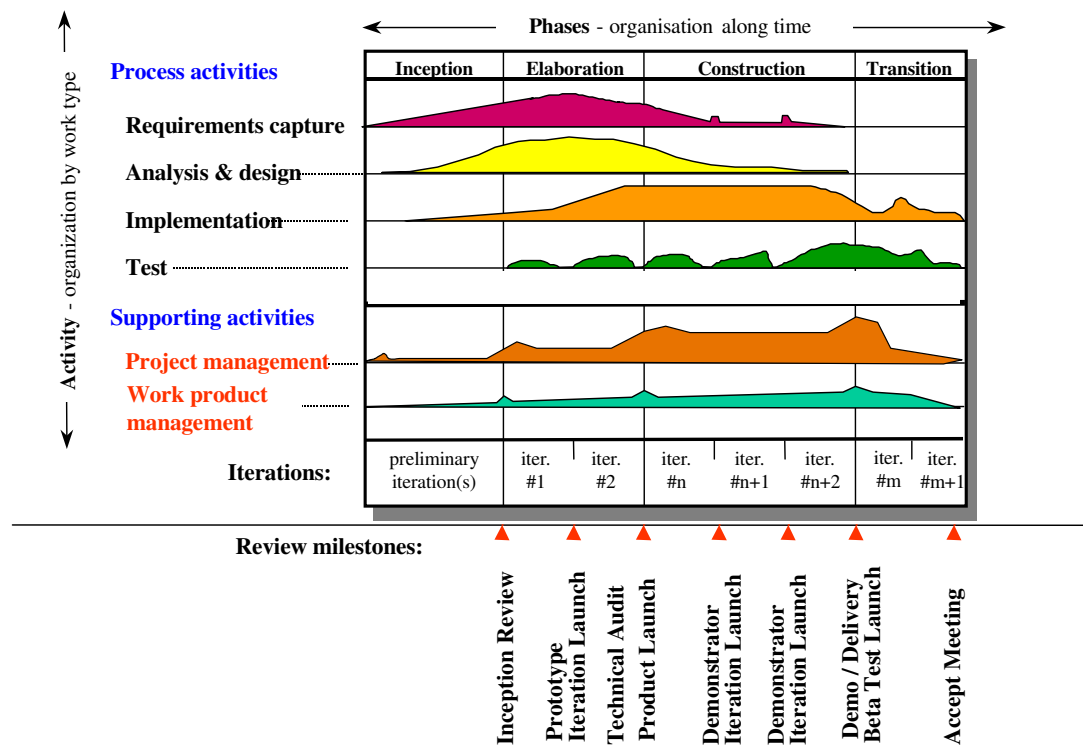
MACCIS rammeverk kommer med en utviklingsprosess for systems engineering som ivaretar følgende aspekter:

- Aktiviteter i utviklingsprosessen
- Iterativ og inkrementell utvikling
- Spesifikasjon av komponentbaserte systemer
- Arbeidspakkehåndtering
- Prosjektledelse

I det videre vil den grunnleggende utviklingsprosess i MACCIS for fremtakelse av komponentbaserte systemer beskrives.

6.2.2.2 Prosessaktiviteter

Aktiviteten i den grunnleggende utviklingsprosessen vises i Figur 73. Figuren viser hovedaktiviteten i et utviklingsprosjekt over tid. Denne figuren er fundamentert og svært likt Unified Process sin modell^[Unif Pr].



Figur 73 Aktivitetene i systemutviklingsprosessen

Prosessen består av fire faser: Startfase, Utdypningsfase, Konstruksjonsfase og Transisjonsfase.

En fase avsluttes når produktet under utvikling har nådd en gitt grad av komplettethet, dermed har vi en milepæl. Milepælene er angitt i underkant av figuren.

Aktivitetene kan deles i to typer: prosessaktiviteter og støtteaktiviteter. Prosessaktiviteter er de aktiviteter som er direkte knyttet til systemutviklingen. Dette er kravinnhenting, analyse og design, implementering og test.

Støtteaktiviteter er de aktiviteter som støtter systemutviklingen for å sikre at disse utføres hensiktsmessig og effektivt. Dette er prosjektledelse og håndtering av arbeidspakker og er videre utledet i kapittel 6.2.2.6

Figur 73 illustrerer godt at i de tidlige faser er det analyse og overordnet arkitektur som dominerer, mens det i senere faser er design på implementasjonsnivå, implementering og test som tar over.

Dette er angitt at kurvens høyre angir mengde av den enkelte aktivitet over tid. Denne figuren er basert på det generiske eksempel, for enkelte prosjekter kan for eksempel krav og arkitektur være vanskelig å ferdigstille tidlig og følge gjennom helt til prosjektslutt. I den grad det finnes standardprosjekter, vil det der normalt være slik at disse aktivitetene er ferdige etter de to første fasene.

6.2.2.3 Iterativ og inkrementell utvikling

Systemutviklingsprosessen følger et iterativt og inkrementelt mønster, et mønster som i dag er akseptert som den foretrukne måte å organisere software utviklingsprosjekter.

Jeg har tidligere i denne oppgaven omtalt den klassiske vannfallsmodellen. Denne modellen har alvorlige svakheter for produktutvikling i en dynamisk kontekst den mest vanlige situasjonen for software tunge utviklingsprosjekter. Her er noen eksempler:

Kravmodellen fryses etter analysefasen. Dermed er det svært lite rom for å justere kursen underveis. Den eneste tilbakekobling fra brukerne før det endelige system er laget er gjennom tekniske dokumenter, noe som fungerer dårlig. En iterativ og inkrementell prosess søker å omgå disse problemene ved å splitte utviklingen opp i mindre deler, der hver del avsluttes med en ufullstendig med brukbar applikasjon som kan testes og forbedres til neste leveranse.

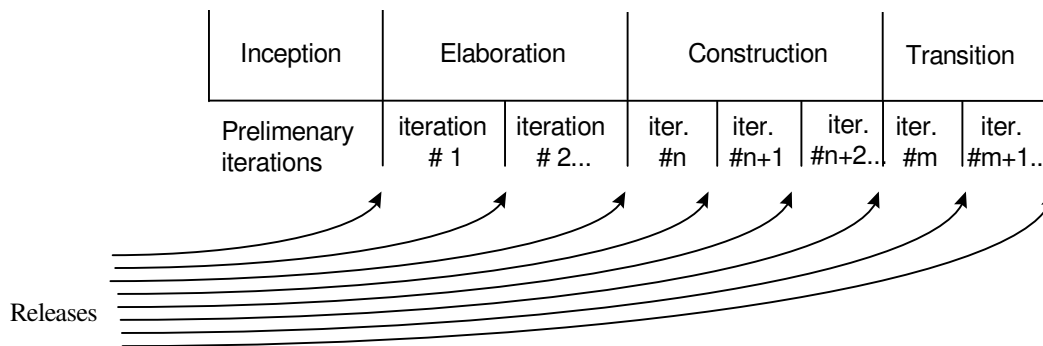
Iterative prosessmodeller deler utviklingen opp i iterasjoner. En iterasjon tester og videreutvikler (deler) av systemet, og dette gjentas over flere iterasjoner. Dette fortsetter til et aksepterbart (sub) system er på plass. En slik oppdagende tilnærming til systemutvikling passer bra når det ikke tidligere er laget tilsvarende systemer. Medaljens bakside kan sies å være at prosessen er vanskelig å planlegge og å kontrollere.

En inkrementell prosessmodell deler systemet opp i komponenter, som igjen implementeres og integreres komponent for komponent. En slik prosess blir da lettere å planlegge og å kontrollere, gitt at utviklerne tidlig får innsikt og etablerer en riktig komponentstruktur.

Denne avhengigheten av å tidlig stabilisere komponentstrukturen er faktisk den største ulempen med en slik tilnærming.

MACCIS systemutviklingsprosess er en iterativ og inkrementell prosess, hvor initielle krav etableres først i form av en USE-casemodell. Deretter designes systemet og implementeres skritt for skritt, hvert steg basert på et eller flere USE-cases. Hvert steg bør resultere i et brukbart system, som kan testes og tilgjengeliggjøres for potensielle brukere. På denne måten er det mulig å videreutvikle USE-casemodellen og dermed gjøre om på resultatene fra forrige fase.

Figur 74 viser hvordan det hvert steg legges til et nytt inkrement til produktet under utvikling. Slik kan man også evaluere systemet under utvikling, og revidere planen basert på det man har erfart. Slik representerer hvert steg en del av prosessen, der hvert steg gir et inkrementell av det endelig produkt, og hvert steg er en iterativ del av den overordnede prosessen.



Figur 74 Iterasjoner i software utviklingsprosessen

6.2.2.4 Spesifikasjon av et komponentbasert system

En fullstendig spesifikasjon av et komponentbasert system må knytte oppførsel i systemet til en beskrivelse av virksomheten der systemet inngår. Dette innbefatter semantikk både hva angår struktur og oppførsel til businessobjektene i virksomheten så vel som intern struktur og oppførsel til systemet i seg selv.

En virksomhetsbeskrivelse omfatter egenskaper ved virksomhetsdomenet (virksomhetsting og deres interne sammenhenger) men overlater så mye som mulig om hvordan disse egenskapene skal ivaretas. Et eksempel kan være at i et økonomisystem vil en virksomhetsbeskrivelse omfatte en generell modell for bokholderi, regler som gjelder, måter og metoder for som kan være halvautomatiske og om mulig distribuerte aktiviteter etc. For en gitt virksomhet vil modellen i tillegg til virksomhetsprosessen også inkludere roller som kan automatiseres og roller som fortsatt skal håndheves av mennesker.

En tilhørende spesifikasjon av komponentbasert system vil omfatte realiseringen av systemet som utfører de identifiserte automatiserbare rollene. En slik spesifikasjon må adressere følgende atskilte men samtidig gjensidig avhengige forhold: Rollene til systemet må i være spesifisert slik at det er sporbart i designet. Semantikken til systemets beregninger må være spesifisert slik at det er sporbart mellom rollene og systemets funksjoner. Strukturen til applikasjonene må være spesifisert som en konfigurasjon av komponenter og samspillet de imellom.

Infrastrukturen i systemet må være spesifisert som tjenester som understøtter applikasjonene.

Kvalitetsmålene til systemet (ytelse, pålitelighet etc.) må være definert og knyttes til de software og hardware komponenter systemet består av. Det er disse komponentene som utgjør systemet, og er basis for systemforvaltning.

For ytterligere informasjon om spesifikasjon av komponentbaserte systemer vises det til Referanse modellen for Open Distributed Processing (RM-ODP).

6.2.2.5 Håndtering av artefakter (work products)

Håndtering av artefakter og arbeidspakkene, vil si håndtering og vedlikehold av alle gjenbrukbare elementer, verifikasjon og godkjenning av deres godhet samt god tilgjengeliggjøring.

Denne aktiviteten går gjennom hele prosjektets levetid, og stopper heller ikke etter overføringsfasen. Figur 73 illustrer arbeidsmengden under prosjektets gang.

I prosjektløpet innebærer denne aktiviteten også å støtte alle aktører som er involvert i gjenbruk av artefakter, håndtere tilbakemeldinger på disse samt å legge inn nye artefakter etter hvert som de produseres. Resultatet av denne aktiviteten er en omfattende katalog av gjenbrukbare ressurser, herunder modeller og kjørbare programmer.

6.2.2.5.1 Hensikt

Håndtering av artefakter skal fremme gjenbruk og er dermed grunnleggende for modell og komponentbasert systemutvikling. MACCIS systemutviklingsprosess argumenterer for gjenbruk.

Håndteringen av artefakter skal gjennomføres uavhengig av utviklingen i prosjektet. De gjenbrukbare enhetene, være seg virksomhetsmodeller, systemmodeller, kravmodeller, designmodeller eller kjørbare programmer, må lagres og tilgjengeliggjøres slik at de er lett tilgjengelige for gjenbruk i hele utviklingsprosjektet.

6.2.2.5.2 Ansvar

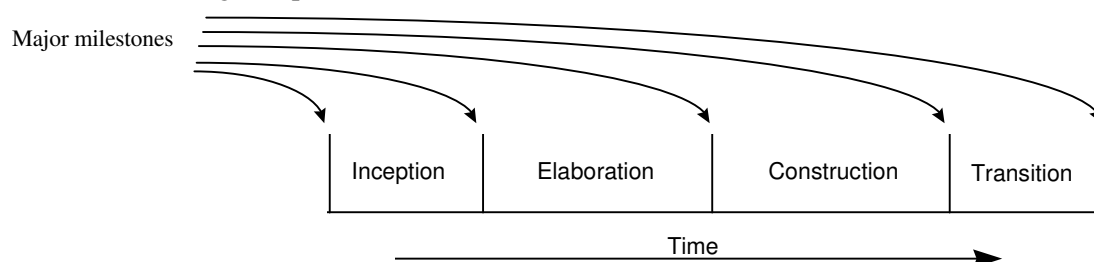
Et eget team må etableres for å håndtere artefakter. Avhengig av prosjektets størrelse kan teamet være fra to personer på deltid til et større team. I tillegg til den viktige jobben som gjøres av dette teamet, er det den etablerte kulturen blant utviklerne for å levere gjenbrukbare komponenter som er suksessen til denne aktiviteten. Alle aktører involvert i utviklingsmodeller og artefakter er ansvarlig for å benytte gjenbrukbare elementer når det er mulig. De er også ansvarlige for å foreslå nye gjenbrukbare elementer for å legge inn i katalogen, samt å forbedre de allerede eksisterende elementer.

Erfaring fra andre prosjekter har vist at håndtering av artefakter bør atskilles fra utviklingsprosjektet og dermed hvis mulig utføres av dedikert personell. I alle tilfeller bør det være dedikerte roller i et prosjekt, med eget budsjett og ressurser. Utviklingsprosjektet bør ikke belemres med å lage generell gjenbrukbare komponenter, rammeverk o.a. Beslutningen om at elementer skal gjøres gjenbrukbare, å lage de gjenbrukbare samt å håndtere katalogen bør gjøres av de rollene som har ansvaret for prosessen håndtering av artefakter.

6.2.2.6 Prosjektledelse

Den andre støtteaktiviteten til systemutviklingen er prosjektledelse. Prosjektledelse er også der den grunnleggende struktur og ansvar for gjennomføringen av prosjektet etableres. Dette innebærer etableringen av faser og milepæler, en plan for iterasjoner samt definisjon av den enkelte iterasjon.

6.2.2.6.1 Faser og milepæler



Figur 75 Fasene i software livssyklus

De fire fasene i software livssyklusen er:

- Startfase (Inception), hvor målsetting i prosjektet defineres.
- Utdypningsfase (Elaboration), hvor problemområdet utforskes, egenkaper spesifiseres, arkitektur designes og resten av prosjektet planlegges
- Konstruksjonsfase (Construction), hvor systemet bygges.
- Overføringsfase (Transition), hvor systemet overføres til brukermiljøet.

Ferdigstillelsen av hver fase er en milepæl i software utviklingsprosessen. Startfasen avsluttes med en gjennomgang, hvor prosjektledelsen må godkjenne videreføring av prosjektet. Et klargjort system avslutter utdypningsfasen. Igjen må prosjektledelsen avgjøre en videreføring, For å avgjøre om systemet er klargjort bør en teknisk gjennomgang (audit) utføres. Her bør man diskutere arkitekturmessig design og tekniske beslutninger, dette kan med fordel gjennomføres med teknisk ekspertise fra utenfor prosjektet selv.

I konstruksjonsfasen gjennomføres en serie med demonstrasjoner ved slutten av hver iterasjon, som angitt i Figur 73. Ved slutten av konstruksjonsfasen avgjøres det om produktet er klart for brukerne og kostnader knyttet til test avgjøres. Overføringsfasen avsluttes med en aksept fra sluttbruker av at systemet er klart for bruk.

6.2.2.6.2 En iterasjon

Hver iterasjon i prosjektets livssyklus kan sees som et lite miniprojekt i seg selv, der alle prosessaktivitetene inngår.

Selv om det er sterke koblinger mellom aktivitetene der den ene avhenger av resultatet fra en tidligere, er det ikke anbefalt å ordne alt i rekkefølge slik den klassiske vannfallsmodellen tilsier.

MACCIS systemutviklingsprosess forskriver parallell utvikling av produkter som henger sammen. Hvis bare utvikleren er klar over hvilke elementer som arbeides på, anbefales det gjerne at man skifter mellom de forskjellige aktivitetene ettersom det anses formålstjenlig.

6.2.2.6.3 Iterasjonsplan

Iterasjonsplanen er grunnlaget for en inkrementell og iterativ utviklingsprosess. Iterasjonsplanen lages i utdypningsfasen. Dette gjøres ved å dele kravene inn i passe deler, og avgjøre i hvilken rekkefølge de skal realiseres. Her anbefales det at iterasjonsplanen er risikodrevet. Ved å følge en risikodrevet strategi, vil de USE-cases med høyest risiko realiseres i første iterasjon.

En klar fordel med en slik inkrementell iterativ prosess er muligheten for tidlig tilbakekobling. Skal dette lykkes må iterasjonsplanen revurderes jevnlig etter hver iterasjon, og om nødvendig endre som følge av erfaringer gjort i siste iterasjon.

Figur 73 viser at prosjektledelsesaktiviteten ikke utgjør mye i starten hvor færre er involvert. Knyttet til den enkelte milepæl er det topper i aktiviteten, så som forberedelser til møter, utarbeidelse av rapporter, etablering av nye planer og iverksetting av disse.

6.2.3 MODELLDREVET ANSKAFFELSE

I min oppgave er det vektlagt anskaffelse fremfor utvikling, da jeg anser dette som mest relevant for den organisasjon jeg arbeider i. Jeg har tidligere vist at de fleste prosessene innenfor software

systemutvikling har fokus på utvikling. Hvordan forholder dette seg så til anskaffelse. I det følgende vil jeg forsøke å angi noen elementer som bør inngå i en modelldrevet anskaffelse.

6.2.3.1 Hensikt

En systemanskaffelsesprosess er en serie med steg og et samspill mellom involverte aktører fra start, gjennom spesifikasjon, bygging og initiell bruk av systemet.

Her anbefales en modellbasert, arkitekturdrevet og evolusjonær systemanskaffelsesprosess – hvor aktørene samhandler tidlig og inkrementelt, gjerne gjennom utvikling, analyse, simulering og kjøring av modeller.

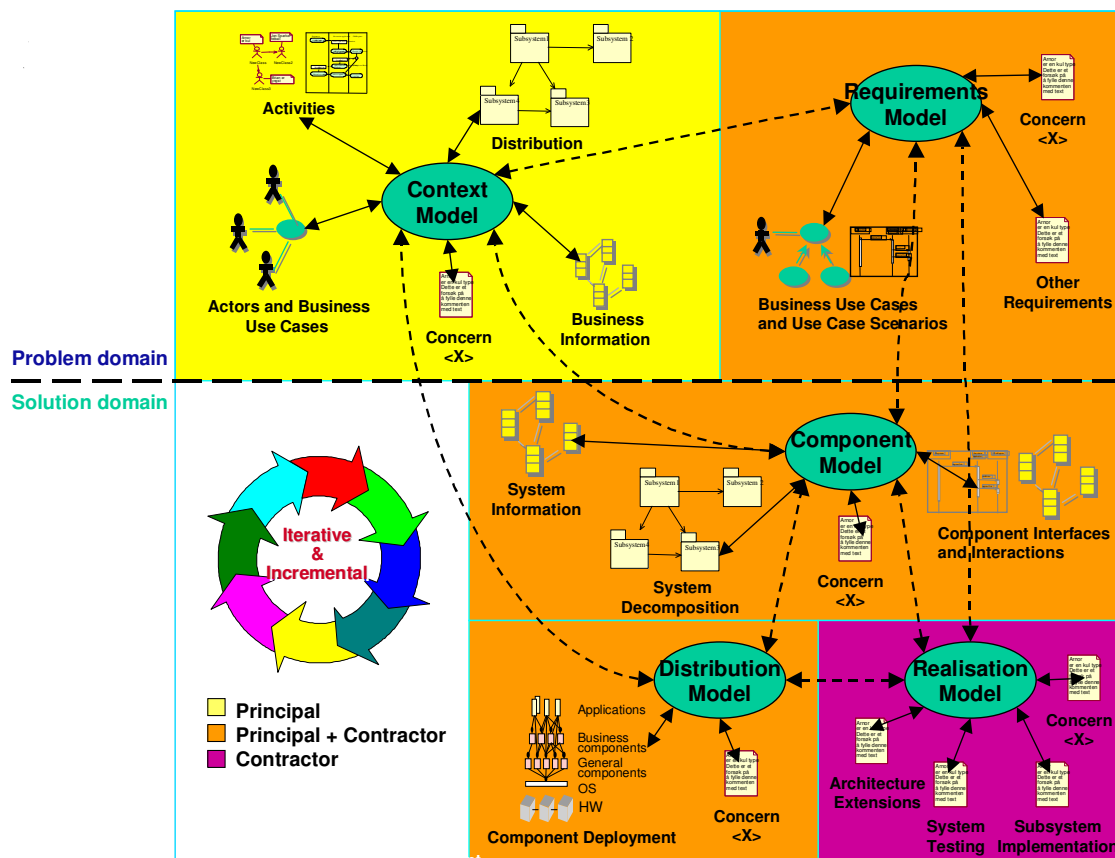
6.2.3.2 Grunnlag

Systemanskaffelsesprosessen kobles til de fire fasene i systemutviklingen, nemlig startfasen, utdypningsfasen, konstruksjonsfasen og overføringsfasen.

Et K2IS er gjerne spesifisert av en hovedaktør der utviklingen settes bort gjennom en kontrakt til en eller flere leverandører.

- En hovedaktør er den som ansetter eller leier inn andre aktører til å levere et realisert system eller delsystem.
- Leverandør er den som inngår kontrakt om å levere et realisert system eller et sett med delsystemer.
-

Figur 76 angir hvilke deler av modellene i MACCIS som utvikles av enten hovedaktøren og/eller leverandøren(e)

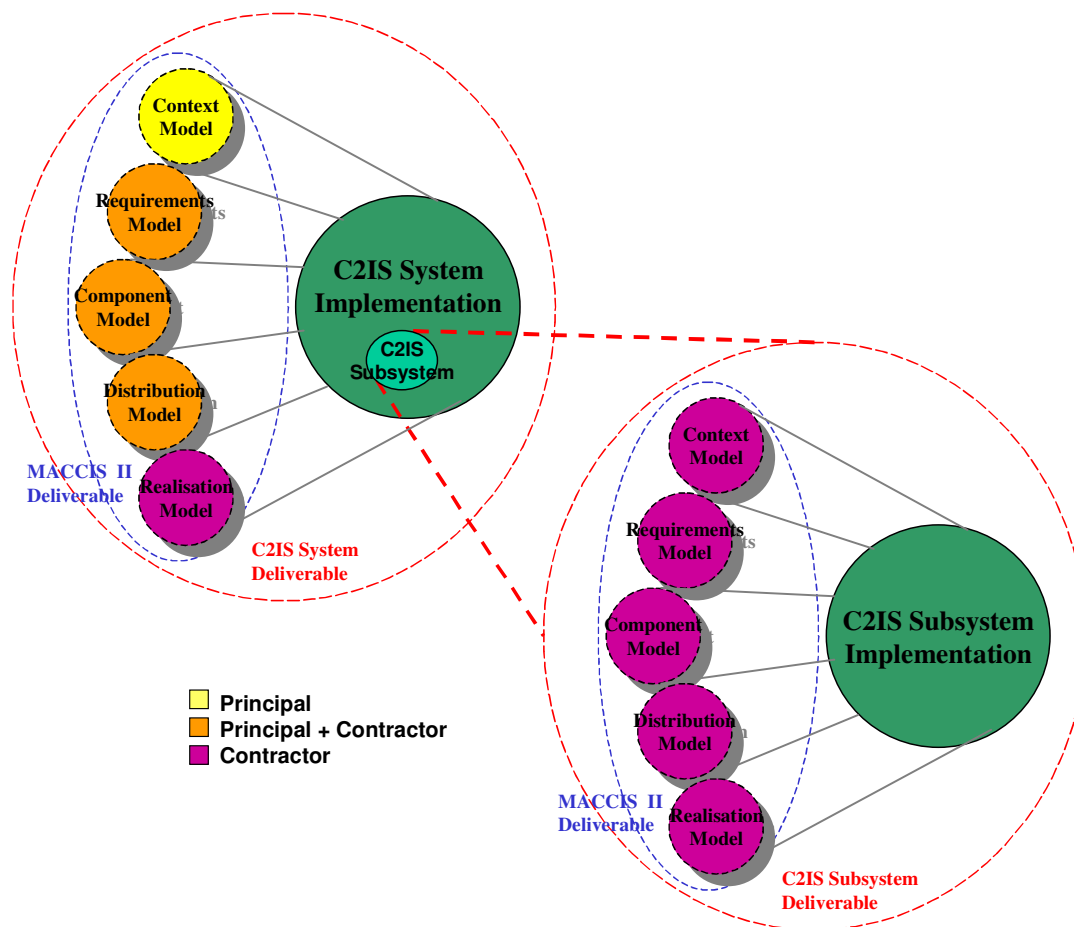


Figur 76 MACCIS prosessoversikt

Figuren angir at kontekstmodellen bør utarbeides av hovedaktøren, mens realiseringsmodellene bør utarbeides av leverandøren.

Spesifikasjonen av systemet bør utarbeides i samarbeid mellom hovedaktøren og leverandøren.

En hovedaktør kan inngå avtale med flere leverandører som hver leverer deler av systemet. Ideelt vil den enkelt komponents arkitektur beskrives ved hjelp av modeller i MACCIS rammeverket, se Figur 77.

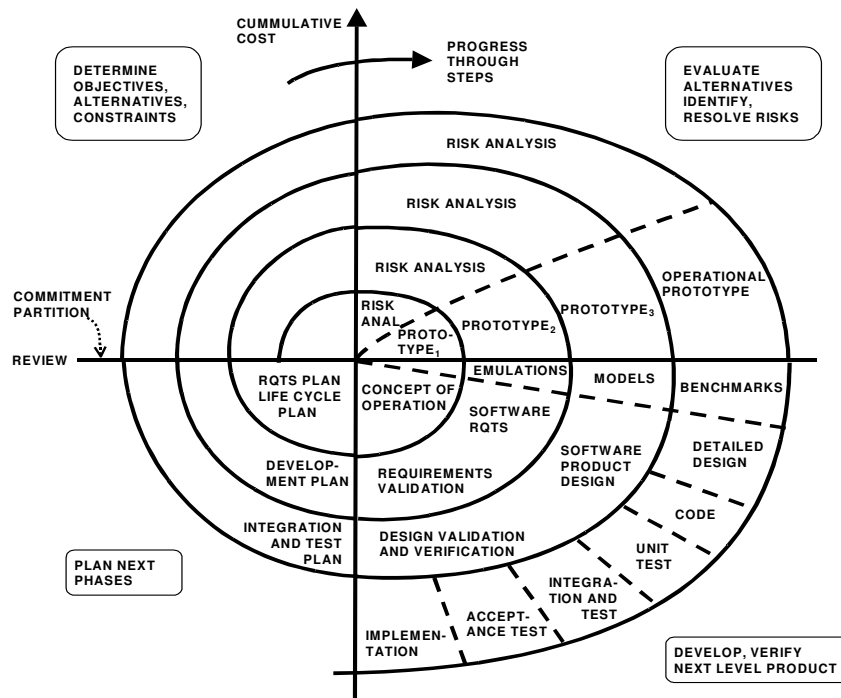


Figur 77 Realisering av K2IS Software Subsystemer

Her anbefales en modellbasert, arkitekturdrevet og evolusjonær systemanskaffelsesprosess – hvor aktørene samhandler tidlig og inkrementelt, gjerne gjennom utvikling, analyse, simulering og kjøring av modeller.

Det anbefales at man følger en stegvis utvikling som angitt i Spiralmodellen og nærmere beskrevet i prosessrammevert som Unified Process.

Ethvert system (uavhengig av nivå) vil alltid være en del av en kontekst, det er derfor avgjørende kontekstmodellen på plass tidlig, og senere anvende denne aktivt i anskaffelsesprosessen.



Figur 78 Livssyklus – Spiralmodellen

Her anbefales en modellbasert, arkitekturdrevet og evolusjonær systemanskaffelsesprosess. Spiralmodellen anbefaler her at dette gjøres ved å fokusere på Life Cycle Objectives (LCO), Life Cycle Architecture (LCA) og Internal Operational Capability (IOC), se Tabell 10 Spiralmodellens metodikk, milepæler og faser.

Milestone Element	Life Cycle Objectives (LCO)	Life Cycle Architecture (LCA)
Definition of Operational Concept	<ul style="list-style-type: none"> • Top-level system objectives and scope - System boundary - Environment parameters and assumptions - Evolution parameters • Operational concept - Operations and maintenance scenarios and parameters - Organizational life cycle responsibilities (stakeholders) 	<ul style="list-style-type: none"> • Elaboration of system objectives and scope of increment • Elaboration of operational concept by increment
System Prototype(s)	<ul style="list-style-type: none"> • Exercise key usage scenarios • Resolve critical risks 	<ul style="list-style-type: none"> • Exercise range of usage scenarios • Resolve major outstanding risks
Definition of System Requirements	<ul style="list-style-type: none"> • Top-level functions, interfaces, quality attribute levels, including: <ul style="list-style-type: none"> - Growth vectors and priorities - Prototypes • Stakeholders' concurrence on essentials 	<ul style="list-style-type: none"> • Elaboration of functions, interfaces, quality attributes, and prototypes by increment - Identification of TBD's (to-be-determined items) • Stakeholders' concurrence on their priority concerns
Definition of System and Software Architecture	<ul style="list-style-type: none"> • Top-level definition of at least one feasible architecture - Physical and logical elements and relationships - Choices of COTS and reusable software elements • Identification of infeasible architecture options 	<ul style="list-style-type: none"> • Choice of architecture and elaboration by increment - Physical and logical components, connectors, configurations, constraints - COTS, reuse choices - Domain architecture and architectural style choices • Architecture evolution parameters
Definition of Life-Cycle Plan	<ul style="list-style-type: none"> • Identification of lifecycle stakeholders <ul style="list-style-type: none"> - Users, customers, developers, maintainers, interoperators, general public, others • Identification of lifecycle process model <ul style="list-style-type: none"> - Top-level stages, increments • Top-level WWWWWH* by stage 	<ul style="list-style-type: none"> • Elaboration of WWWWWH* for Initial Operational Capability (IOC) - Partial elaboration, identification of key TBD's for later increments
Feasibility Rationale	<ul style="list-style-type: none"> • Assurance of consistency among elements above <ul style="list-style-type: none"> - via analysis, measurement, prototyping, simulation, etc. - Business case analysis for requirements, feasible architectures 	<ul style="list-style-type: none"> • Assurance of consistency among elements above • All major risks resolved or covered by risk management plan

*WWWWWH: Why, What, When, Who, Where, How, How Much

Tabell 10 Spiralmodellens metodikk, milepæler og faser

- Life Cycle Objectives (LCO):

Eiere og sentrale aktørers (Stakeholders)' engasjement til støtte for arkitekturarbeide (Forlovelse)

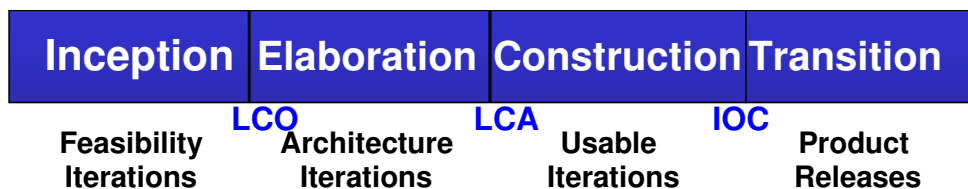
- Life Cycle Architecture (LCA)

Eiere og sentrale aktørers (Stakeholders) engasjement til å støtte en full livssyklus (Inngå giftemål)

- Initial Operational Capability (IOC)

Eiere og sentrale aktørers (Stakeholders) engasjement til å støtte operativ bruk (Som å få det første barnet!)

Begrepene LCO, LCA og IOC er såpass utbredt, at jeg ikke har ansett det nyttig å oversette disse. Begrepene inngår i prosessrammeverk så som Unified Process, se Figur 79.

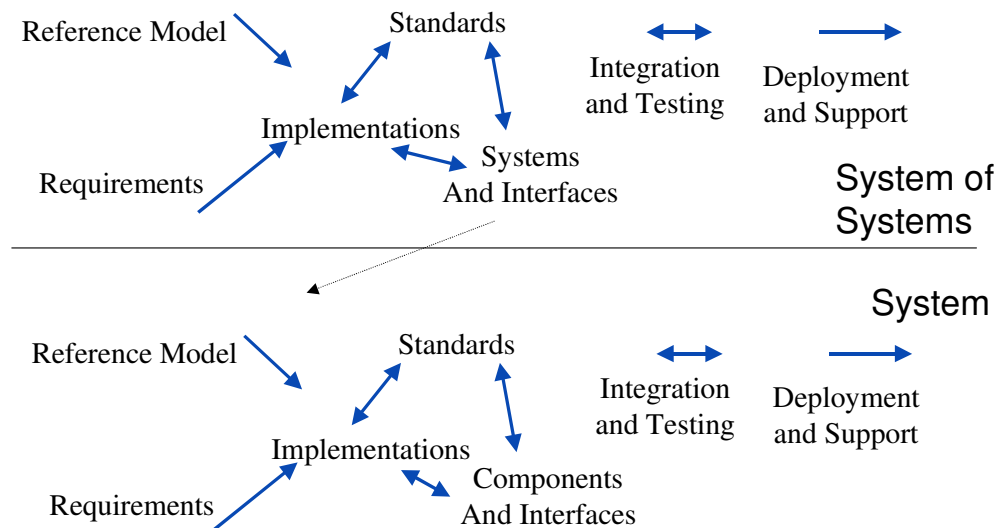


Figur 79 Unified Process og LCO, LCA og IOC

Systemanskaffelse bør fokusere på nødvendig systemutvikling. Når det eksisterer allerede utviklede systemer, komponenter eller standarder bør disse benyttes.

En analyse av et potensielt (sub) system bør gjøres ved hjelp av en relevant referansem modell. Basert på en forståelse av kravene kan man utvikle forskjellige arkitekturbaserte alternativer hvor potensielle

standarder vurderes. Her vil kvaliteten på denne analysen avhenge av at mulige leverandører kan delta i denne analysen. Eventuelle kontraktuelle problemer knyttet til å involvere leverandører tidlig må derfor håndteres. En systemmodell er det tidligere vist i denne oppgaven, består av et hierarki der systemer og systemer av systemer inngår i et større system. Konstruksjon av et system ut fra allerede eksisterende systemer, fordrer at man har brukbare modeller og standarder, og at man fokuserer på integrasjon og testing.



Figur 80 System of Systems med gjenbruk av eksisterende Implementasjoner

7 SYSTEMANSKAFFELSE

7.1 INNLEDNING

Ideen med kapitlet er å beskrive hvordan man bør gjennomføre anskaffelser av informasjonssystemer over hele livssyklusen.

Jeg innfører her to elementer i tillegg til MACCIS. MACCIS anser jeg som en svært god støtte til å gjennomføre informasjonssystemutvikling også anskaffelse. Basert på egne erfaringer anser jeg at MACCIS ikke kan innføres i en organisasjon som min egen uten å ha en viss kontroll på to andre elementer: beslutningsprosessen og læring eller modenhet i egen organisasjon.

Slik jeg ser det kan MACCIS innføres gradvis hvis det er vilje og evne til å etablere systemforvaltning med utgangspunkt i MACCIS. Videre må beslutninger som organisasjonen tar være bygget på arkitekturelementer fra MACCIS. Dette innebærer at man etablerer en referansearkitektur etter mønster fra MACCIS.

Leverandører som skal levere til Forsvaret har en etablert utviklingsprosess, her antas det at de benytter RUP.

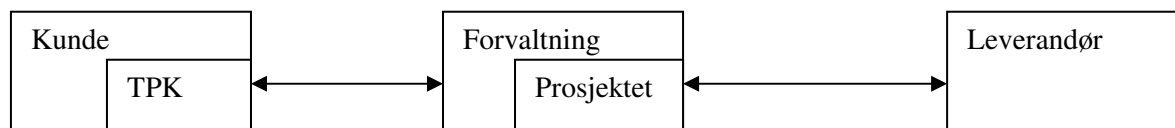
Utfordringene knyttet til anskaffelse av større systemer er delvis beskrevet i kapittel 2. Forsvaret har sine metoder for anskaffelse, PRINSIX og ARF, disse er beskrevet i kapittel 4.6.

Systemarbeid har slik jeg har erfart det på Institutt for Informatikk vært fokusert rundt det å selv skulle skape et system, altså sett fra et leverandørperspektiv, enten som en intern utviklingsorganisasjon eller som en ekstern leverandør. Kundeperspektivet er forsøkt ivaretatt gjennom "brukermedvirkning". Brukeren forsøkes brakt inn i utviklingsprosjektet. Til tross for at dette i dag er utbredt i mange utviklingsprosjekter, og i høyeste grad velmenende finner jeg dette ikke er dekkende for Hæren som en stor organisasjon. Kundens rolle slik jeg ser det, når Hæren er kunde er mye mer enn at vi skal være med å påvirke produksjonen av et nytt IT-system. Vi har et noe brokete kundebegrep. Jeg vil derfor starte med å avklare dette noe nærmere.

Mange av Forsvarets kontrakter involverer flere store leverandører. I tillegg til at det er merkantile utfordringer har dette også noen ganger tendens til å bli industripolitisk spørsmål. Faglige sett ligger det merkantile utenfor systemarbeid. Det politiske aspekt berører jeg ikke i det hele tatt. Jeg har derfor ikke tenkt at dette kapitlet skal være uttømmende om alle de fallgruver som ligger innen dette området. Jeg vil liste opp de problemområder jeg ser, og muligens skissere noen løsninger.

For egen organisasjon er kundebegrepet alltid en utfordring. Forsvaret har derfor definert en rolle, Totalprosjektkoordinator TPK som koordinerer kunden.

Samspillet mellom kunde, prosjekt og leverandør er komplekst. Når vi vet at leverandørrollen sjelden eller aldri er en, men en sammensetning av leverandører er den forenklete modellen – en ideell fremstilling av virkeligheten.



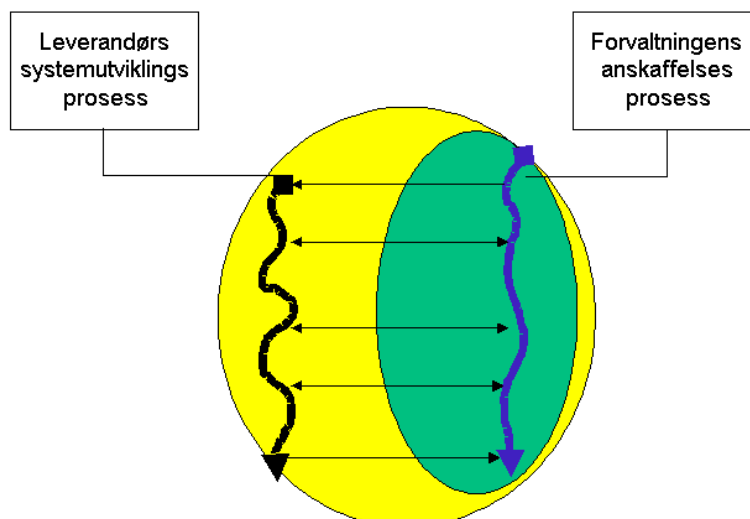
Figur 81 Begrepsapparat

7.2 EKSISTERENDE METODERS ANVENDELSE

Metoder for anskaffelse av informasjonssystemer, hvorfor er det interessant? Jeg vil kort oppsummere med at jeg ikke kan se at vi i Forsvaret ikke har dette i dag.

7.2.1.1 Innledning

Metoder for utvikling av informasjonssystemer er beskrevet i kapittel 4.1.4. Det eksisterer flere metoder på markedet i dag. Det er naturlig at man skjeler til den erfaring som er gjort på systemutvikling når man skal se på anskaffelse av nye systemer. Det er likevel ikke gitt at man kan kopiere dette – da anskaffelsen skal forholde seg til resultatene av utviklingsprosessen – ikke hele prosessen som sådan. En ting er leverandørens prosess, en annen ting er forvaltningens prosess, en tredje ting er hvordan disse to forholder seg til hverandre. Figur 82 Leverandør - forvaltningsprosess illustrerer denne tredelingen. Hvis man ønsker å styre denne helheten må man fra forvaltningens side ha en definert og forstått prosess. Videre kan man kjenne leverandørens prosess, men dette er ikke alltid mulig. Det man må ha et forhold til er hvordan man skal samvirke med leverandøren. Vi antar at leverandøren har en definert prosess, som vi i vårt tilfelle antar er RUP. Samvirke mellom egen prosess som kunde i forhold til en leverandør som bruker RUP har jeg ikke funnet noe litteratur om.



Figur 82 Leverandør - forvaltningsprosess

7.2.1.2 Metoder for anskaffelse av systemer

Det eksisterer i dag ingen godt etablert metode for anskaffelse av informasjonssystemer. Med utgangspunkt i de eksisterende metodene for systemutvikling som finnes, kan man velge en tilnærming. Hvis man har kunnskap om hvilke aktiviteter som skal gjennomføres kan man lage seg en tilpasset modell, og deretter velge å sette ut noen aktiviteter til leverandør. De aktiviteter man ikke kan eller vil sette ut, må man da gjennomføre selv. Hvordan man finner fram til hvilke aktiviteter en anskaffelsesprosess bør, kan eller skal sette ut til utviklingsprosessen har jeg ikke funnet litteratur om. Dette er derfor slik jeg ser det et forholdsvis ubeskrevet område.

7.2.1.3 Hvorfor egne metoder for anskaffelse av Informasjonssystemer?

Spørsmålet kan stilles, all den tid anskaffelser gjøres daglig på alle områder. Hva er forskjellen mellom å kjøpe et hus, en bil en stridsvogn og et informasjonssystem?

Jeg tar en kjensgjerning for gitt – den er at å anskaffe og vedlikeholde informasjonssystemer er noe mer komplekst enn å anskaffe andre tekniske systemer. Hovedgrunnen til dette er den nærhet informasjonssystemet har til de mellommenneskelige prosessene i organisasjonen. Det er kanskje større likheter med ”organisasjonssystemer” enn med mekaniske systemer. Jeg mener at dette i dag er erkjent innenfor utvikling av informasjonssystemer. Dette innebærer at IT-industrien med utstrakt bruk at konsulenter har en videre forståelse for kompleksiteten enn det mange kunder har.

Oppslag om feilslåtte IT-prosjekter i pressen, f eks TRESS-90 kan muligens illustrere behovet for andre metoder for anskaffelse av IT-systemer.

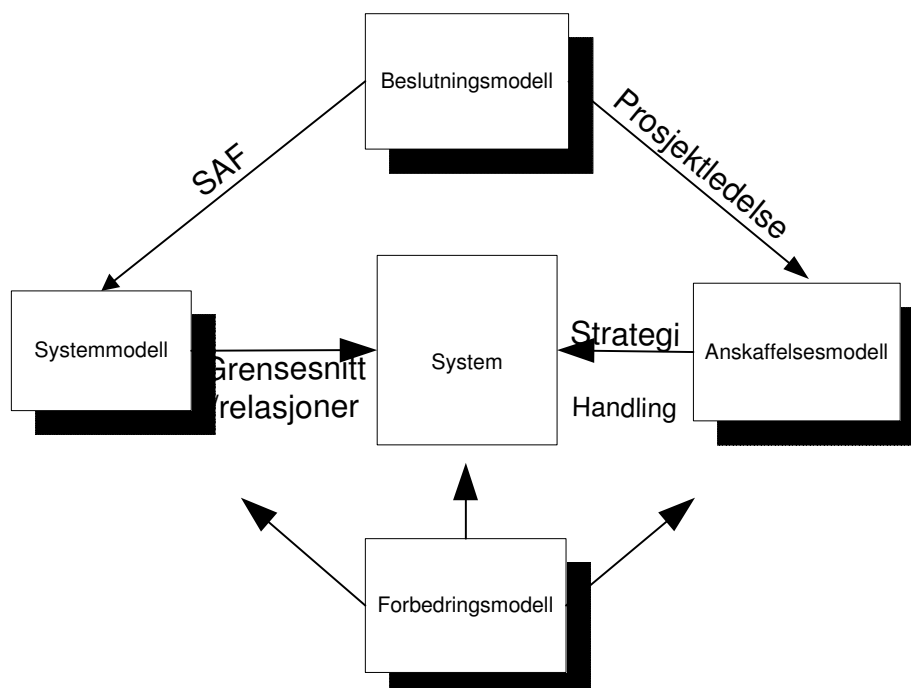
Gjeldende metode for anskaffelse i Forsvaret PRINSIX, fremmer en vannfallsorientert tankegang. Dette er ikke en anbefalt metode for anskaffelse av informasjonssystemer.

7.3 MODELLAPPARAT

Med utgangspunkt i tidligere kapittel 2 har jeg gjort meg noen erfaringer under arbeidet med prosjektgjennomføringen. Det følgende kapittel ser på eksisterende metoders anvendelse, i lys av kapittel 2 og de erfaringer som jeg har gjort som medarbeider i et prosjekt.

Deretter presenteres en pakke med modeller: Beslutningsmodell, Systemmodell, Anskaffelsesmodell og Forbedringsmodell.

Modellene kan stå hver for seg som verktøy for den enkelte prosessen, men har også en innbyrdes relasjon se Figur 83.



Figur 83 Modellapparat – systemanskaffelse

De enkelte modellene presenteres i de følgende kapitlene.

7.3.1 MODELLAPPARAT – EN HELHET

En slik portefølje av modeller kan virke forvirrende. Det jeg forsøker å gjøre er sette systemet i sentrum. Systemet er det alt dreier seg om, enten det skal anskaffes, endres eller forvaltes. Videre ser jeg mange aktiviteter som knytter seg til det å håndtere systemet. Jeg forsøker å gruppere disse aktivitetene i pakker – en modell som pakker inn en gruppe aktiviteter.

7.3.1.1 Beslutningsmodell

Hvilke beslutninger er så vesentlige at de ikke alene kan tas av prosjektlederen? Hvordan forholder prosjektlederen seg til andre interessenter?

Disse spørsmål forsøkes samlet i begrepet beslutningsmodell.

7.3.1.2 Systemmodell

MACCIS som tidligere er beskrevet har en serie med metoder og modeller for å håndtere kompleksitet. Her gjøres et utdrag av de vesentlige elementene som har betydning for de andre modellene, og dette kaller jeg systemmodell.

7.3.1.3 Anskaffelsesmodell

Hvordan skal man gjennomføre anskaffelse av informasjonssystemer? Hvordan sikrer man at man iverksetter de riktige aktivitetene, de aktivitetene som både gir kundetilfredshet og som har økonomisk bærekraft over tid? MACCIS som tidligere er beskrevet har drøftet de vesentlige elementene.

Dette oppsummeres her og jeg kaller det anskaffelsesmodell.

7.3.1.4 Forbedringsmodell

En erfaring jeg har fått etter å ha jobbet med anskaffelse av informasjonssystemer gjennom de siste 8 år, er at vi får ikke til alt første gang. Har vi evne til å lære samtidig som vi gjennomfører anskaffelser? En vanlig påstand er: "Vi kan jo vente med å dokumentere dette til vi er ferdige!" Jeg mener det har vist seg mange ganger nå at dette aldri ble gjort. Slik jeg ser det er det kontinuerlig evne til å lære over tid som er den viktigste suksessfaktoren – derfor har jeg gitt forbedringsmodellen en sentral plass i mitt modellapparat.

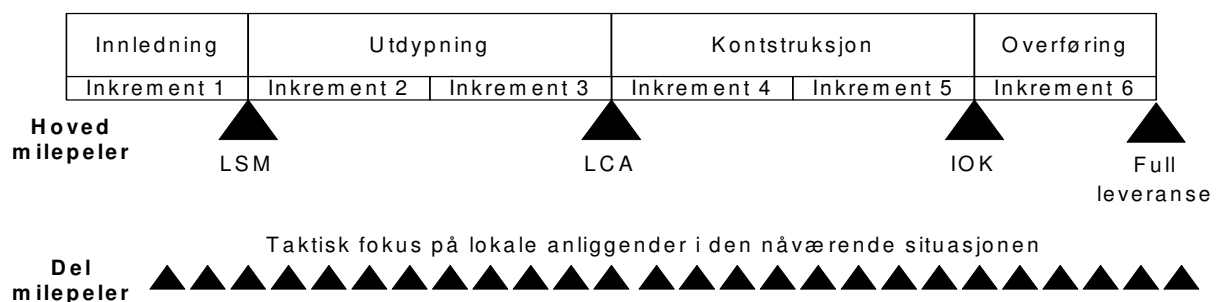
7.3.2 BESLUTNINGSMODELL

Nye argumenter for å bruke mer penger tar aldri slutt. Dette skyldes endringer som skjer i livssyklusen til vårt software system, og som krever at beslutningsunderlaget lages om og om igjen.

Beslutninger har slik jeg har opplevd det en tendens til å bli fattet på svært varierende grunnlag. Fra beslutninger som kommer som resultat av lange studier som har veid for og imot til de daglige beslutninger som tas uten noe underlag. Dette er nok vanlig i de fleste organisasjoner. Slik jeg ser det mangler vi en felles forståelse for hvordan man skal underbygge en beslutning. Jeg har derfor valgt å lete etter noe som kan hjelpe oss her.

Det jeg har funnet er en modell som støtter den kontinuerlige prosessen å fremme underlag for beslutninger. Modellen, er i store trekk basert på boken "Making the Software Business Case – Improvement by the Numbers" av Donald J. Reifer. Reifer kaller modellen for et rammeverk, og kaller denne for "Model-Based (System) Architecting and Software Engineering – MBASE". Dette rammeverket er basert på et arbeid ved University of Southern California, som hadde som målsetting å lage et rammeverk for å drive livssyklus planlegging og vurderinger uavhengig av andre modeller. (Her refereres det ikke uventet til Rational Unified Process – RUP [Kruchten, 1988], Spiral [Boehm, 1988] osv)

Modellen har fire vesentlige milepæler som er Livssyklus målsetting (LSM), Livssyklus Arkitektur (LSA), Initiell operativ kapasitet (IOK) og Full leveranse. Modellen åpner for at arbeidet kan gjøres sekvensielt eller iterativt innenfor hver fase.



Figur 84 MBASE Milepæler²⁴

I innledningsfasen lages budsjetter, og disse oppdateres etter hvert. Bruk av penger fordeles over hele perioden også budsjettene vil endre seg. Dette skyldes at kravene endrer seg ettersom vi lærer mer om hva kunden virkelig vil ha eller trenger. En usikkerhetsavsetning er ikke uvanlig for å ivareta dette. En dreven software ingeniør vet også at dette er stedet å gå og spørre om penger – gitt at man har en god begrunnelse. Det er derfor på tide å se på hvordan man kan komme opp med gode argumenter – gode beslutningsunderlag. Vi skal også grave dypere inn i den prosessen med å lage gode forretningsmodeller (developing business cases) Dette skal vi gjøre ved å knytte software utviklingsprosessen og ledelsesprosessen sammen gjennom hele livssyklusen. Verktøyet som foreslås brukt er Goal - Question – Metric (GQM)

7.3.2.1 Hvordan sette mål ved bruk av GQM

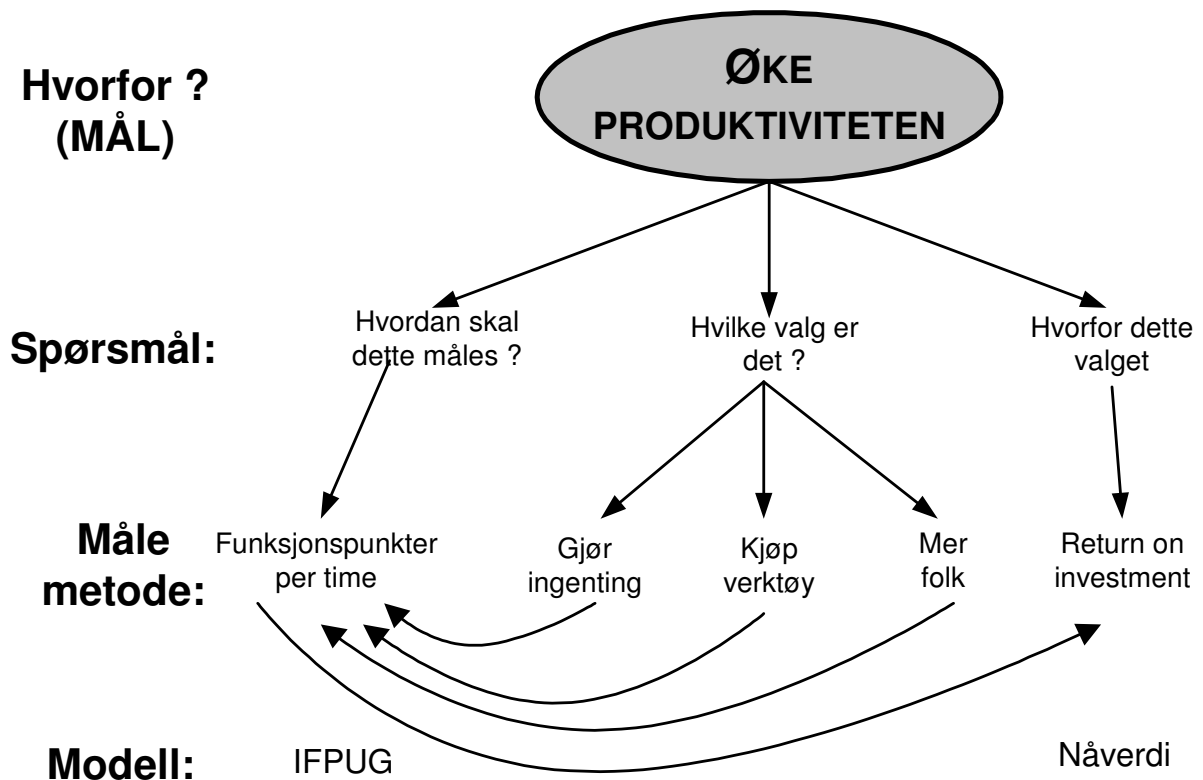
Når det er spørsmål om endringer eller forbedringer er det naturlig å spørre "hvorfor". Min erfaring er at det ofte blir oppfattet som svært negativt å stille dette spørsmålet. Men det burde være mulig å svare på et slikt spørsmål – og på den måten få klarhet i hvordan endringen hjelper oss i forhold til vår organisasjons mål. Skulle vi lage et eksempel på dette tar vi utgangspunkt i de organisatoriske mål. Hvis man ikke har noe spesifikt kan nok forbedringsfirkanten brukes i de fleste tilfeller. Her er spørsmålet, øke produktiviteten, redusere leveransetid, kutte kostnader eller forbedre kvaliteten.

²⁴ Kilde: Figuren er tatt fra Boehm, B. W., "MBASE Tutorial presentation at the Los Angeles SPIN meeting 2001

Reduser	Unngå/Kutt
Leveransetid	Kostnad
Produktivitet	Kvalitet
Øke	Forbedre

Figur 85 Forbedringsfirkanten

Hvis vi tar et eksempel som brukes i boken av Reifer – Hvorfor kjøpe et nytt verktøy? En uformell tilnærming som da ofte fører frem er denne. Du må da finne det spørsmålet som fører frem til den målemetoden og den modellen som underbygger ditt svar. Hvorfor kjøpe et verktøy? Hvilke valg har man? Hva er kostnaden og nytten? Hvilken måte skal bruke til å måle denne kostnad og en eventuell nytte. Forslaget er å utlede en beslutningsmodell ved hjelp av logikken i GQM.



Figur 86 GQM eksempel

Den GQM informasjon som samles inn gir retningslinjer og guider deg gjennom hvordan du skal underbygge anskaffelsen av verktøy. Faktisk gir figuren svar på flere ting med et øyekast:

Målet er å øke produktiviteten

For eksempel kan produktivitet måles vha Funksjonspunkter per time (International Function Point User's Groups ^[IFPUG] brukes som referanse)

Tre alternativer er vurdert: ikke gjøre noe, kjøpe verktøyet eller tilføre mer personell

Du underbygger valget av den beste løsningen ved å bruke "return of investment – ROI" ved å bruke nåverdi av investeringen.

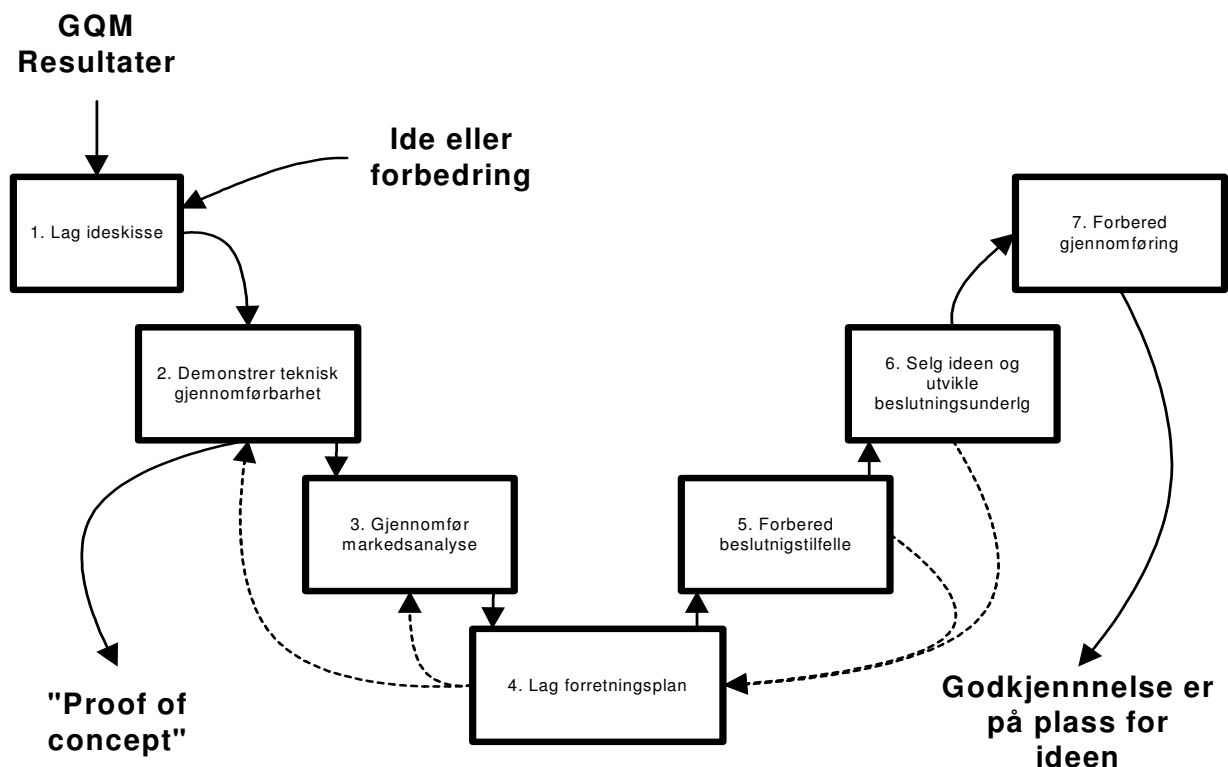
Nåverdiberegning er ikke akkurat mitt fagfelt – men er så langt jeg kan forstå en måte å bruke dagens verdi av pengene som grunnlag for å fatte en forstående beslutning om bruk av penger. Dette virker fornuftig, og bør derfor vurderes som en teknikk.

7.3.2.2 Hvordan komme i gang?

Hensikten med en "business case" – eller et beslutningstilfelle er å fremskaffe et godt og overbevisende underlag til ledelsen for å underbygge en ide eller et forslag til forbedringer. Beslutningstilfellet er i så måte vesentlig fordi dette gir økonomiske, konkurransemessige eller på andre måter fundamentet for å bruke penger, talent eller tid på ditt forslag fremfor det noen andre kan

finne på å fremme. En misforståelse som ofte oppstår her – det at man har laget et beslutningsunderlag en gang betyr ikke at man ikke må gjøre dette på nytt. Beslutningstilfeller er en del av software anskaffelsesprosjektet og hele livssyklusen.

Vel verdt å huske er at det som skal foreslås må være teknisk så vel som økonomisk attraktivt. I dag er det vanskelig å få til en investering i teknologi – bare fordi det er spennende. Det er vel verdt å finne et overordnet mål. Reifer nevner naturlig nok profitt som overordnet mål. I vårt tilfelle i forvaltningen vil det å redusere kostnader. Dette er vel så godt som profitt. Selv i de tilfeller der det ikke eksisterer budsjetter for den aktivitet som foreslås – bare det er økonomisk godt nok fundert vil midlene kunne dukke opp. Alt dreier seg om å finne argumenter for å overbevise ledelsen om at investeringen holder vann. Her kommer relasjonen til de overordnede mål – og dermed bruk av GQM inn. Dette betyr valg og vurderinger av mulige løsninger så vel som det å bygge en sunn anbefaling. En faktor som kan være avgjørende er om ledelsen er opptatt av det du vil ha en beslutning om. Så det kan godt sies at for å få til en satsing er form og presentasjon vel så viktig som innhold. I boken til Reifer omtales beslutningstilfeller som forbedringsforslag. Reifer kommenterer her at de fleste forbedringsforslag stoppes eller hindres av kulturell og sosial motstand mot endring. Dette bør derfor tas med i betraktning når man skal gjennomføre en forbedring må man ta i betraktning denne innebygde motstand som er tilstede i de fleste organisasjoner. Dette må bakes inn som en del den plan som lages. Ved å knytte behovet for endring til de overordnede mål vil ledelsen få de argumentene som trengs. GQM, fremholder Reifer gir startpunktet og rammen for å lage et godt beslutningstilfelle. Med utgangspunkt i GQM kan man sette i gang en syvstegs forretningsplan prosess, se Figur 87.



Figur 87 Sjustegs forretningsplanprosess

Reifer skisserer denne modellen brukt i et software utviklingsprosjekt. For et anskaffelsesprosjekt mener jeg hovedideen kunne brukes. Naturlig nok vil det være noen endringer – de kommenteres i den enkelte trinn av prosessen.

7.3.2.2.1 Lag idéskisse

En idéskisse (Reifer bruker begrepet "White Paper") er et viktig første steg i prosessen. Her utformes idéen klart og argumentene for hvorfor dette er viktig for vår organisasjon trer frem. Kort sagt kan man kalle dette et salgspapir – hvis det virker slik at beslutningstakere fenger på idéen.

Hvis det ikke kan uttrykkes på noen få sider (Reifer sier mindre enn 10 sider) – så la være! Bruk bilder og grafikk for å underbygge budskapet. Dokumentet skal være så spesifikk som overhodet mulig, det skal være fengende og godt skrevet. Det viktigste som skal frem er handlinger som skal gjennomføres for å få det beskevene resultat. De kulturelle endringene skal komme frem i dagen og diskuteres.

7.3.2.2.2 Demonstrer teknisk gjennomførbarhet

Hvis det ikke allerede er gjort, skal det gjennomføres en demonstrasjon av anvendbarheten som neste steg. Demonstrasjoner kan fungere som bevisførsel ved at andre kan "se og ta på" idéen. For å få dette i land uten at det skal ta all verdens tid og ressurser, er det vesentlig å vise frem det essensielle. Dette skal ikke være noen fullstendig softwareløsning. Det skal være avgrenset til å vise hvordan man håndterer de risiki som de fleste vil spørre etter. Hvis beslutningstakere kan "se og ta på" idéen vil de lettere forstå hva som ønskes oppnådd.

I et anskaffelsesprosjekt må man enten ha egne ressurser som evner å lage disse enklere demonstratorene. Hvis ikke kan man bruke utvalgte leverandører for å lage dette. Dette krever at man har en viss evne til å kjøpe slike tjenester – det må være budsjettert for.

7.3.2.2.3 Gjennomfør markeds- og innslagsanalyse

Her er det en klar forskjell mellom utviklings- og anskaffelsesprosjektet. Jeg mener vi kan benytte det samme steget – men med "motsatt fortegn". Her kan anskaffelsesprosjektet vurdere tilstedeværelse av ideen i løsninger i markedet. I tillegg må vi analysere hvordan denne nye ideen passer inn i vår arkitektur.

7.3.2.2.3.1 Markedsanalyse

Finnes løsningen hos flere leverandører, vil risiko for å gjennomføre dette være håndterbar. Finnes den ikke vil dette være en klar indikator på behov for større årvåkenhet.

7.3.2.2.3.2 Innslagsanalyse

Her vurderes hvordan den nye ideen passer inn i eksisterende arkitektur. Hvis tilsvarende ideer er brukt før vil gjenbruk av grensesnitt sannsynligvis være mulig. Er ideen slik at den krever helt nye løsninger, må dette gjøres kjent slik at dette tas med som en oppgave for gjennomføringen av ideen.

7.3.2.2.4 Lag gjennomføringsplan

(Reifer kaller dette forretningsplan) I et anskaffelsesprosjekt vil det være andre ting som vektlegges enn i et utviklingsprosjekt. De fleste organisasjoner trenger en forretningsplan før man får penger til en ide. En gjennomføringsplan for oss kan lages etter mal fra PRINSIX' gjennomførbarhetsstudie. Det å lage gode gjennomførbarhetsstudier som også inkluderer behov for ressurser er slik jeg kjenner forvaltningen i dag ikke prosjektenes sterkeste side. Det kan ikke fortsette at man får gjennomslag for ideer som krever ressurser som man ikke har, ei heller har tenkt å skaffe seg.

7.3.2.2.5 Forbered beslutningstilfelle

Her er muligheten for å forbedre underlaget fra idéskissen. Som del av planen som nå er så å si ferdig, lager vi en forbedret underlag for "salg" av idéen. Her bør det vektlegges finansielle og eller risikorelaterte argumenter.

Dette er i praksis en kortversjon av planen – en powerpointpresentasjon.

7.3.2.2.6 Forsalg av ideen og lag beslutningsunderlag

Det å lage planen er ikke nok. Planen må selges – et forsalg legger grunnlaget for et godt salg. Støtte kan komme kommer fra alle deler av organisasjonen. For å få gjennomslag trengs støtte fra noen som har gjennomslagskraft. En sponsor – en som sitter riktig plassert, trengs for å vinne gjennom de politiske og økonomiske diskusjonene som nå kommer. Riktig bruk av demonstrator og et godt forberedt beslutningstilfelle er ofte de beste argumentene.

7.3.2.2.7 Forbered gjennomføring

Som siste steg må man være klar til å sette i gang når beslutningen er tatt. Dette betyr at man har utført de delene som alltid tar lengst tid og som er viktigst for å få ideen ut i praksis.

Med gjennomføringsplanen på plass – overfør dette til prosjektplanen. Sett milepæler og finn kritisk vei. Gjør de praktiske håndgrepene som trengs for å få tilgang til de økonomiske midlene som stilles til disposisjon Finn de riktige ressursene og få tilgang til disse så snart arbeidet krever det. Stillingsbeskrivelser for de jobbene som skal gjennomføres må være klare på forhånd.

Identifiser de du mener du kan spille på i organisasjonen. Din sponsor bør sitte i styringskomiteen. Organisasjonskartet må kunne forberedes med de som nå skal være en del denne jobben Utstyr og verktøy som trengs må være på plass. Finn et egnet arbeidssted som gjør at gruppen som nå skal jobbe sammen kan være lokalisert hensiktsmessig i forhold til hverandre.

7.3.2.3 Prosessen i praksis

Den beskrevne prosessen virker mildt sagt noe overveldende. Reifer sier: "Du kan nok lure på når du får bruk for denne prosessen. Det ser ut som det er mye arbeide"

Jeg har ikke gått i detalj på hvor hensiktsmessig alle stegene vil være for vår organisasjon. En tilpasning – der de hendelsene som ikke direkte overførbare til "vår verden" fjernes er nok på sin plass. Jeg mener at elementer av denne prosessen bør kunne nyttes også i vår organisasjon. Slik jeg ser det vil dette klarere få frem beslutningsunderlaget for de beslutningene som vi må leve med etter store endringer. Den beskrevne prosess er nok mest egnet til å gjennomføre de store beslutningene. Dette er beslutninger som har behov for mange ressurser og stor kulturell endring.

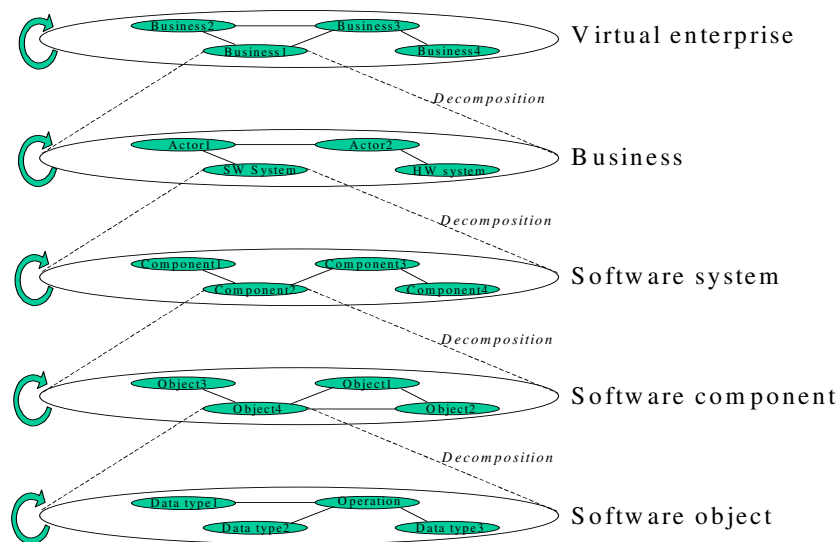
Proessen som er beskrevet har elementer som vil være anvendelig i mindre skala – og som vil kunne brukes gjennom hele livssyklusen til software anskaffelsesprosjektet

7.3.2.4 Beslutningstilfelle og software livssyklus

Deler av den beskrevne prosess for beslutningstilfeller vil brukes gjennom software anskaffelsesprosjektet. Dette kan være behov for endringer og forbedringer. Hvis GQM brukes aktivt vil dette fungere som en viktig brikke i å ta de riktige beslutninger underveis i prosjektet.

7.3.3 SYSTEMMODELL

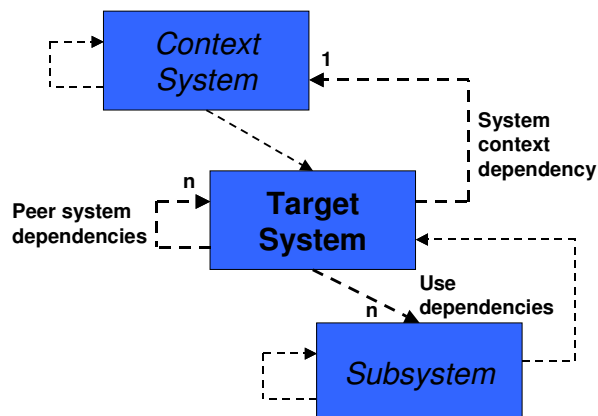
For å snakke om systemanskaffelse er det første som bør avklares: hva er "systemet". MACCIS anfører at system kan brukes om selvstendige enheter på mange nivå. Dette oppfattes som et hierarki fra et overordnet system, eller en "virtuell organisasjon" via virksomhetsnivå med flere samspillende enheter til softwaresystemet – som er vårt primære anliggende. Softwaresystemet igjen kan som kjent brytes ned i ytterligere enheter – disse ønsker vi også skal være selvstendige enheter som kan utvikles og forvaltes over tid. Figur 88 illustrerer dette hierarkiet av systemer – der vi nå setter softwaresystemet i sentrum.



Figur 88 Nivåer i systemperspektivet

7.3.3.1 Hva er systemet?

Vårt definisjonsområde kan ikke bare avgrenses til hardware og software alene. Vi erkjenner at mennesker, deres behov overfor vårt system er en del av de krav som settes til systemet. Videre vil systemet vårt bestå av enheter som har egne egenskaper. Figur 88 illustrerer hvordan systemet vårt kan sees som en del av det overliggende system – omgivelsen og de underliggende system – subsystem.

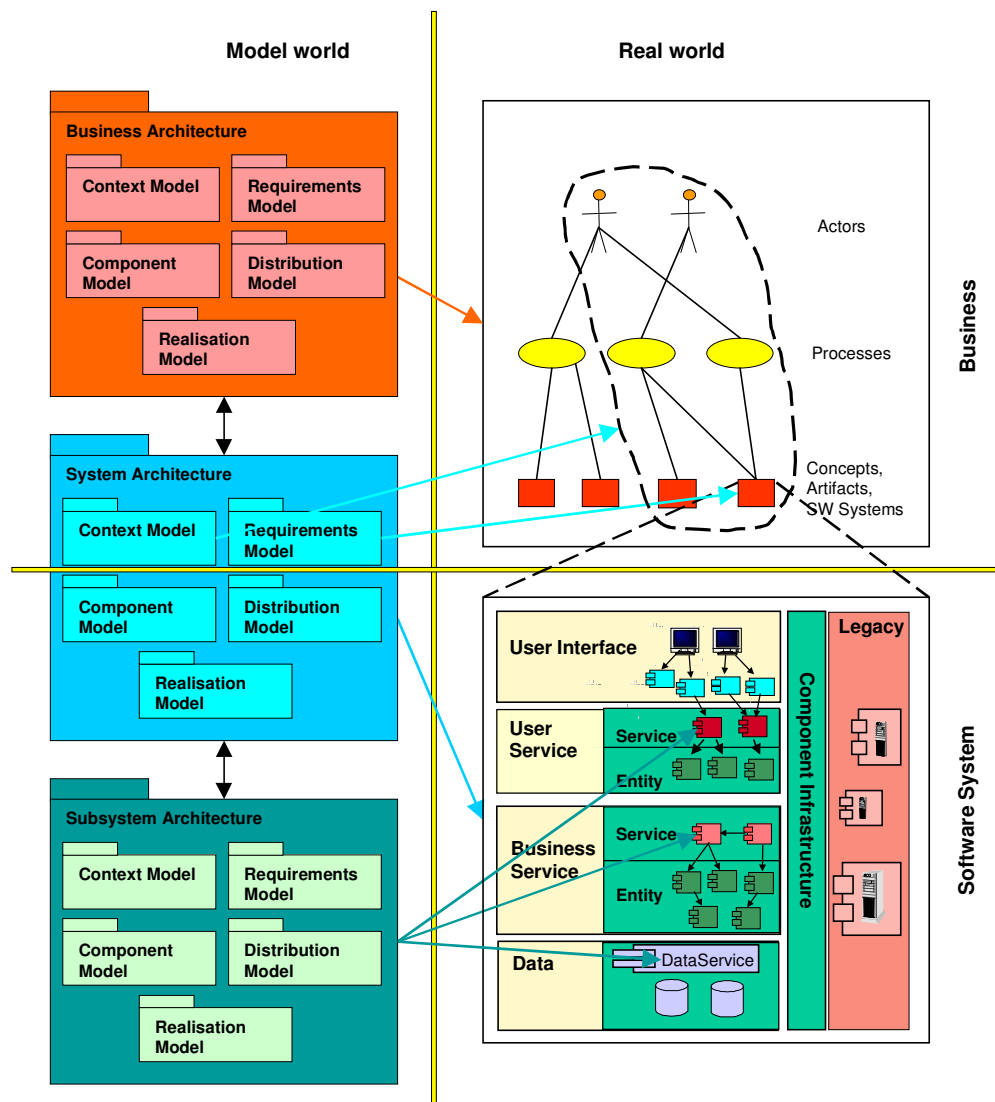


Figur 89 Systemhierarki - generisk modell

Begrepet "Target system" – målsystemet, er det systemet som skal utvikles eller forvaltes "over tid". Begrepet "Context system" – omgivelsessystemet, representerer det høyer ordens system som vårt målsystem skal støtte eller være en del av. Relasjonen "System context dependency" angir denne avhengigheten. Begrepet "Subsystem" – delsystemet, er de komponenter og byggesteiner vi bruker når vi spesifiserer eller bruker under utviklingen av vårt målsystem. Relasjonen "Use dependencies" angir denne avhengigheten. Vårt målsystem har også ofte relasjoner til andre systemer på samme nivå. Relasjonen "Peer system dependencies" angir denne avhengigheten. MACCIS beskriver hvordan hvert nivå sitt system kan beskrives ved å benytte et sett av UML-modeller.

7.3.3.2 Systemets arkitekturmodell

Systemet er ikke avgrenset til kun seg selv – målsystemet. Ved et gitt tidspunkt vil det være ønskelig å avgrense systemet for å kunne spesifiseres et behov for endring. Over tid vil vi ønske å ha en modell over systemet som del av det overordnede system og som bestående av delsystemer. MACCIS - en oversikt over modellarkitekturen illustrerer fokus for systemets arkitekturmodell, og hvordan dette henger sammen med i særdeleshet det overordnede system – virksomhetens arkitekturmodell.



Figur 90 MACCIS - en oversikt over modellarkitekturen

Vår systemmodell er altså et sett med modeller som beskriver, omgivelsen til systemet, krav til systemet, komponenter i systemet, distribusjon av systemet og realisering av systemet. For å håndtere endringene i virksomhet og krav anbefaler vi å lage en modell for det overordnede nivå – en virksomhetsmodell. Hvilke prosesser som skal understøttes og kravene til systemet avledes av denne modellen. For å håndtere valg av delsystemer og endringer på disse over tid anbefaler vi også å lage en modell for hvert system. Hvis dette er delsystemer som lages av andre, vil dette være deres ansvar. Da vil vi kun ha en oppfatning av disse som selvstendige komponenter i vår systemmodell.

7.3.3.3 Hvordan komme i gang?

Når vi blir presenter MACCIS virker ideen svært tiltrekkende. Svært mange blir motivert for å sette i gang med å modellere. Noen er dog mer skeptiske for de ser hvor omfattende dette kan bli. Må man modellere hele virksomheten først? Må alle modellene for systemet på plass og må vi forstå alle modeller og de egenskaper til systemet dette beskriver før vi kan gå til en anskaffelse? Vår kunde sier: ”Jeg vet jo hva jeg vil ha – jeg vil ha et system for å holde situasjonsbildet oppdatert”.

MACCIS er ikke et hinder for å imøtekomme dette ønsket.

Jeg mener det er mulig å starte en anskaffelsesprosess av et målsystem, uten å ha alle modeller for virksomheten på plass. Det er også mulig å sette i gang anskaffelse av delsystemer uten å ha alle modeller for målsystemet på plass. Ved å lage kontekstmodellen for vårt målsystem lager vi også en beskrivelse av en komponent i virksomheten. Hvis vårt system beskrives på en slik måte at denne kan forstås av de som driver med virksomhetsmodellering, vil de kunne gjenbruke våre modeller når de skal bygge sin virksomhet. På samme måte, hvis det eksisterer tekniske løsninger på markedet med beskrevne egenskaper, så kan dette delsystemet beskrives inn som en del av vårt målsystem. Slik jeg ser det er det bevissthet om denne endring som vil skje over tid som er nøkkelen til om man tør å iverksette anskaffelsen av noe som man erkjenner man ikke forstår helhetlig. En målsetting i gjennomføring av anskaffelsen må da være å ta frem systemmodellen, slik at dette blir forstått.

7.3.4 ANSKAFFELSESMODELL

7.3.4.1 Innledning

Basert på de erfaringer jeg har gjort med prosjekt K2IS tillater jeg meg å skissere en ny helhetlig anskaffelsesmodell. Dette er i store trekk en oppsummering av hovedtrekk fra [MACCIS]

7.3.4.1.1 Forutsetninger:

De som skal lage produktet – dette antas å være en leverandør og de - antas å ha en utviklingsmetode. Vi antar at de benytter Rational Unified Process. De som skal stå for gjennomføringen av anskaffelsen - vanligvis omtalt som prosjektet, antas å ha en metode for å gjøre dette, i vår kontekst er dette styrt av gjeldene regler, dette er primært ARF og PRINSIX. Organisasjonen som skal ha det nye produktet, antas å ha noen retningslinjer for forvaltning av eksisterende systemer. Forsvarsdepartementet har i 2004 utgitt nye retningslinjer for materiellforvaltning i Forsvaret for å sikre en helhetlig og strategisk styring av materiellforvaltningen. Disse retningslinjene har ligget til grunn for Forsvarssjefens direktiv for materiellforvaltning. Dette erstatter Tjenestereglement for Forsvaret klasse 7 (TfF kl. 7) materiellforvaltning. ^[St-prp-1]

Mitt forslag er at de i tillegg må utføre dette i tråd med en referansearkitektur, dette vil for oss si at arkitekturprosessen er operasjonalisert. Slik jeg ser det har vi da tre prosesser. Dette er systemutvikling hos leverandør, det er prosjektledelse²⁵, og det er en forvaltningsprosess hos mottakende organisasjon..



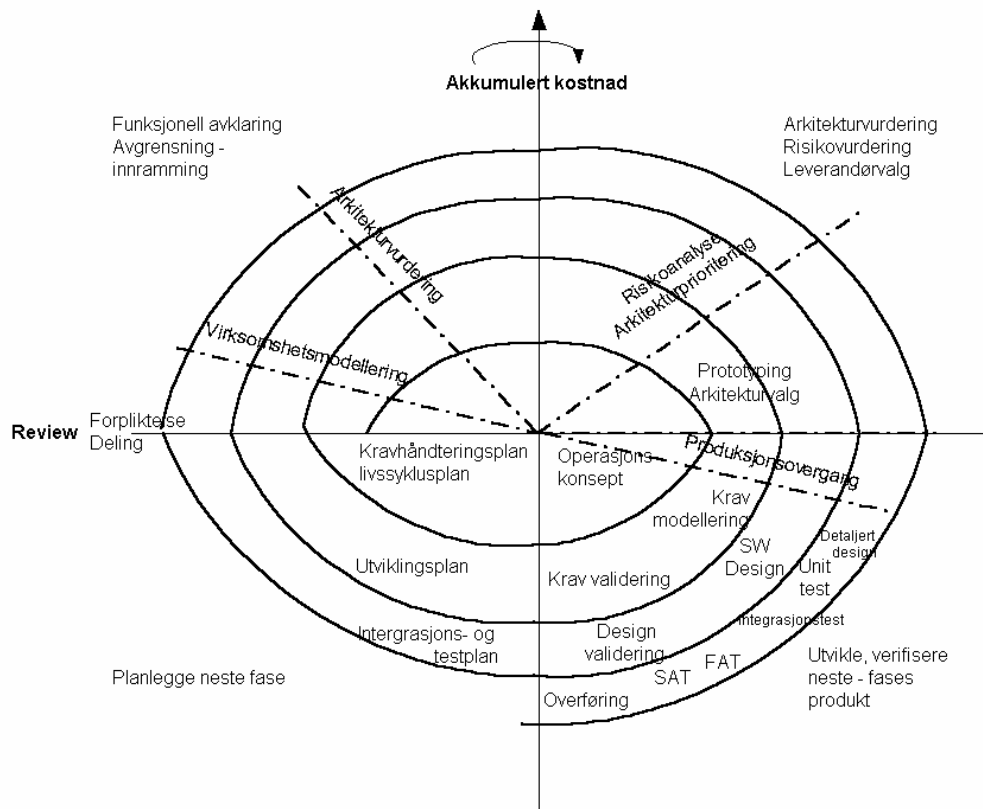
Figur 91 - 3 prosesser

Spørsmål: Hvordan far man disse prosessene til å virke sammen?

7.3.4.1.2 Hypotese:

En anskaffelsesmodell basert på spiralmodellen, drevet av risiko antas å gi en synkronisering, se Figur 92 Anskaffelsesmodell.

²⁵ prosjektet vil sannsynligvis foregå som et samspill mellom de involverte aktører både hos leverandør i FMO



Figur 92 Anskaffelsesmodell

7.3.4.2 Styringselementer for K2IS Blokk X

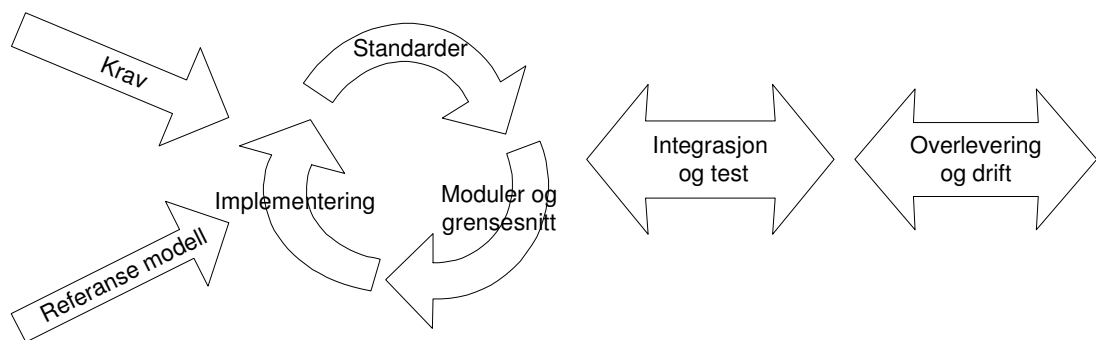
- *Arkitektur ikke arkeologi!*
- *Tidsavgrensete kontrakter – inkrementelle leveranser*
- *Risikoreduksjon*

Hensikten med å anvende denne metoden er å:

- Redusere uklarheter
- Fjerne forsinkelser
- Redusere risiko
- Redusere kostnad
- Øke kvaliteten
- Øke vedlikeholdbarheten
- Øke mottakelighet
- Bibeholde av designmessig/arkitektonisk integritet
- Forbedre integrasjon med arven
- Ivareta opprinnelig integrasjon og økt stabilitet overfor endringer
- Øke pålitelighet av grensesnitt

7.3.4.3 Overordnet prosessmodell

Hvilket bilde kan man bruke som er et rikt bilde på det vi ønsker å oppnå? Dette kan jeg si at jeg har lett ganske lenge for å finne. Det jeg har lett etter må kunne fungere for ”store organisasjoner” og det må kunne fungere ”over tid”. Videre har vi et krav om at dette skal ”gjøres på den for forsvarets billigste måte”. Jeg har funnet en god figur jeg føler griper denne overordnede målsettingen, se Figur 93 Åpne standardbaserte systemer. Jeg har ikke funnet igjen kilden, men denne figuren illustrerer at vi basert på de føringer som er gitt av ”arv” og de nye kravene skal finne løsningen ved hjelp av standarder og moduler og definerte grensesnitt. Implementering er bare en del av denne sirkelen, på lik linje med valg av standarder og definering av moduler og grensesnitt. Jeg skal ikke si at denne figuren forteller at det nødvendigvis er en tredeling mellom disse tre elementene. Men hvis man har problemer med å argumentere for fokus på annet enn implementering kan figuren være en god hjelp. Videre består helheten i figuren av en balanse mellom utviklings- og implementeringsfasen og integrasjon og test samt overføring og drift.



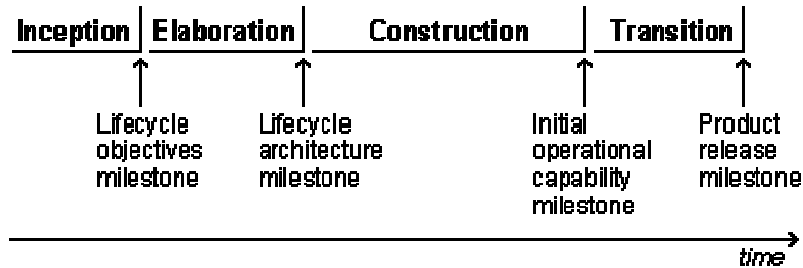
Figur 93 Åpne standardbaserte systemer

7.3.4.4 Faser

Vi må anta at det er mulig å etablere de tre prosessene. Dette er en antagelse som ikke er helt urealistisk. Alle utviklingsorganisasjoner har en eller annen form for prosess. De fleste prosjekter styres etter en eller annen modell. De fleste organisasjoner forvalter sine systemer og gjør prioriteringer for nye prosjekter basert på en eller annen prosess. Hvorvidt disse prosessene er etablerte og eksplisitt beskrevet er en annen sak. Gitt de retningslinjer som legges opp i CMM, se kapittel 0, kan man anta at det er en fordel å ha både etablerte prosedyrer og metoder for å vedlikeholde og kontinuerlig forbedre disse. Basert på egne erfaringer fra prosjekt K2IS tror jeg at det ikke er formålstjenlig å bruke for mye energi på å pådytte en leverandør en metode. Dette er et av de valg en ekstern organisasjon må gjøre selv. Hva man kan gjøre er å gjøre avtale om noen kontrollerende tiltak, gjerne i form av milepæler som synkroniserer den prosessen de har med det kunden forventer.

7.3.4.5 Milepæler

Rational Unified Process benyttes som veiledende modell. I Rational Unified Process har vi fire faser.



Frem til disse fire milepælene har vi følgende hovedaktiviteter:

- Funksjonell prioritering (En trade-off mellom virksomhetskrav)
- Arkitekturleverandørvalg (En trade-off mellom det som er og det som skal komme)
- Risikoreduksjon (Utvikling-leveranse) (En trade off mellom økonomi-fremdrift-kvalitet)
- Evaluering

7.3.4.5.1 Funksjonell prioritering

I Hæren innefor K2IS har vi gjennom mange år tenkt virksomhetsmodellering. Dette er gjort ut fra flere målsettinger. En målsetting er å velge virksomhetsområder hvor informasjonsteknologi skal brukes for å oppnå endringer. En stor organisasjon bør ha en formening om hvilke deler av virksomheten som realiserer hvilke deler av virksomhetens visjoner og mål.

Dette brukes for å prioritere hvilke prosesser, handlinger og funksjoner som skal støttes av informasjonsteknologi. Teknologien skal innføres slik at endringer kan observeres, måles og evalueres. Dette gir igjen innspill til neste inkrement av leveranser.

Følgende handlinger oppsummerer den funksjonelle prioriteringen, se Tabell 11.

Målsetting	Handling
Reduser uklarheter	Utvidet bruk av simulering og prototyper
	Spesifiser datamodell og globale standarder
Håndtere Kravendring	Bruk prototyper som baseline for raskt å oppdage betydning av endringer
Fjerne forsinkelser	Tidlig spesifikasjon av standard grensesnitt for å redusere manipulering av data
Øke kvaliteten	
Øke effektiviteten	
Tids og kostbesparelser	Gjennomføre funksjonalitet – kost avveininger tidlig i kravanalysefasen. Undersøk om små endringer kan gi stor besparelser i tid og kostnad
Øk fleksibiliteten maksimalt	Sett ytelseskrav og funksjonelle krav til helheten – i tråd med den overordnede målsetting.
Minimer omfanget	Skreddersy dokumentasjonskrav til det som trengs av sluttbruker for å forstå, benytte og vedlikeholde software produktet. (reduser mengden av ofte ubenyttet dokumentasjon, tatt frem for å monitorere hvert enkelt utviklingstrinn
Bruk av COTS	Gjennomføre markedsundersøkelser og bedriftsbesøk som en del av prosessen med RFP. Dette gjør at kravene kan tilpasses eksisterende løsninger når de fortsatt er i tråd med den overordnede målsettingen.
	Oppfordre sluttbrukerne slik at de i størst mulig grad benytter kommersielt tilgjengelige funksjoner, i stedet for å fremme sære militære krav
Inkrementelle leveranser	Benytt de mekanismer som er mulig i kontrakten for å fremme inkrementell kravutvikling/definisjon, i stedet for å hamre alle kravene ved oppstart

Tabell 11 Hvordan nå målsettinger i den funksjonelle avklaringen

7.3.4.5.2 Arkitektur - Leverandørvalg

Her vektlegge tre elementer, dette er:

- Arkitektur for å fremme helhetsperspektivet foran delperspektivene!
- Arkitektur for å fremme en systematisk gjennomgang basert på erfaring!
- Arkitektur for å fremme fokus på grensesnitt, gjenbruk og avklare usikkerhet

Disse overordnede målsettinger, gjør at jeg må fremme en betenkning.

Leverandør ønsker ikke nødvendigvis å bidra til at vi får noe mer enn det vi betaler for.

Eksempel COE (Common operating Environment) definert som "gjenbrukbare komponenter". Det hevdes at det koster mer å lage komponenter som kan brukes mange ganger. Er det komponentene vi vil ha, eller den skissen som ligger til grunn for å lage komponenten – modellen av komponenten? Antagelsen er at dette er den samme skissen "hver gang" – og at i den eksplisitte anvendelsen er det bare snakk om en spesialisering. Holder dette vann, vil vi med et riktig sett av modeller kunne spesialisere systemet for flere (alle) anvendelser.

Følgende handlinger vektlegges for å nå denne målsettingen:

Målsetting	Handling
Reduser kostnad	Involvere leverandør i konseptfasen
	Bruk "best value" anskaffelsesteknikk, slik at dette letter avveieg mellom kravoppfyllelse og kostnad.
Reduser risiko	Bruk integrerte team.
COTS	Belønne valg av standardløsninger – kommersielle standarder – standard arkitekturvalg. Gjør valget slik at dette er et målbart seleksjonskriterium.
Åpenhet	Oppfordre til bruk av åpne standarder (open systems) heller enn å kreve at spesielle militære standarder overholde. Dette kan lettere gjennomføres hvis et integrert team er tilstede i hele fasen.
Håndtering av uklare krav	Anskaffelsesmetode (inkrementell!) skal lett kunne håndtere endringer i krav, og dermed kunne fjerne uklare krav.
Kvalitet	Legge fokus på resultatene fra gode kvalitetssystemer hos leverandøren (ISO/AQAP/CMM) Dette legges inn som del av utkast til kontrakt.

Tabell 12 Hvordan nå målsettinger for arkitektur og leverandørvalg

7.3.4.5.3 Risikoreduksjon (prioriteringsmekanisme)

Håndtering av risiko dreier seg mye om å se de store trekkene. Derfor anbefales en grundig forstudie. Videre skal fokus legges på inkrementelle leveranser – noe som virker og kan prøves ut; særlig mot brukerne for å avstemme forventninger og krav. Tidlige leveranser er et kommunikasjonsmiddel mellom bruker og utvikler som fremmer dialog. Prototyper – tre typer; teknisk(risikoreduserende for tekniske valg) gaffel(tar frem en helhet og viser omfanget – brukergrensesnitt) stuprør (viser en funksjonalitet helt gjennom, og tar da for seg alle elementer - alle funksjoner innefor dette området) Hvilken type skal velges? Etter RUP er det risikoreduksjon, kanskje særlig på teknisk usikkerhet, infrastruktur som fremmes (sammen med prototyper som tas frem på MMI, drevet av USE-case)

Kundens krav og handlinger i utviklingen oppsummeres slik:

Målsetting	Handling
Redusere uklarheter	Bruk åpne standarder (open systems standards)
Redusere kostnad	
Øke kvaliteten	
Fjerne forsinkelser	Krev bruk av standard interface, og reduser antallet Engineering genererte endringer. Eliminere "nice to have" og utsett "have to have" til senere leveranser/oppgraderinger
Reduser risiko	Evaluere COTS nøye og krev bruk av standard grensesnitt. Benytt gjerne "virtuelle" grensesnitt for å få bedre klarhet i designet.
Øke kvaliteten	
Standard utviklingsprosess	Legg fokus på leveransene fra utviklingsprosess heller enn å kreve spesiell Rational Rose, Mil Std 498/IEEE 1220 eller en avart av dette.
Inkrementelle leveranser	Bruk dette som fremste teknikk for å redusere risiko. Involver de endelige brukerne slik at de tidlig får et produkt de kan vurdere.
Minimer formalisme	Prosjektledelsen fokuserer på planer og status heller enn å se på unødige detaljerte designpresentasjoner. Delegere dette til mindre formelle forum, og la implementasjon gjennomføres ved at brukerne deltar i integrerte utviklingsteam.
Reduser vedlikeholdskostnad	Oppfordre leverandøren til å kommersialisere produktet as is, eller med små endringer. Dette kan med fordel gjøre så tidlig som mulig.
Gjenbruk	Gjør det lønnsomt for leverandøren å gjøre gjenbruk av software bibliotek Benytt gjerne "clearing house" ²⁶ for å lette tilgangen på hvilke kommersielle produkter som er tilgjengelige og tilstrekkelige for de som skal anskaffe.

Tabell 13 Hvordan nå målsettinger for risikoreduksjon

7.3.4.5.4 Evaluering

Hva har vi lært? Læringen går på to nivåer – produkt og prosess. Hva har vi lært om produktet i denne fasen, slik at vi bedre kan beskrive det som en del av et større hele. Hva har vi lært om prosessen, hva fungerte bra og hvordan kan vi forbedre de tingene som ikke fungerte bra?

²⁶clearing House – anses å være en mekanisme for informasjonsdeling [A **clearing house** (also spelled *clearing-house* or *clearinghouse*) is an institution that collects and distributes information, <http://en.wikipedia.org/wiki/Clearing-house>, søk gjort 2006-11-13.]

Dette kan oppsummeres slik:

Målsetting	Handling
Produktbeskrivelse	Har vi en beskrivelse vi nå kan leve med?
Test og verifikasjon	Har vi fått det vi ba om? Hva må gjøres om?
Prosessforbedring	Hva må beskrives bedre i neste fase? Hvilke beskrivelse ble ikke brukt?

Tabell 14 Hvordan nå målsettinger for evaluering

7.3.5 MODENHETSMODELL FOR ANSKAFFELESORGANISASJONEN

7.3.5.1 Innledning

Alle virksomheter burde ha et behov for å måle modenhet til deres interne anskaffelsesprosess for software^[SA-CMM]. Hensikten med å måle modenhet av organisasjonens anskaffelsesprosess for software er å identifisere områder hvor forbedringer er nødvendig.

For å kunne gjøre forbedringer, må de kjenne det endelige mål og hva som trengs for å nå dette målet. I tillegg er det nødvendig å kunne måle fremgangen mot målet.

En dugelighets- og modenhetsmodell gir det nødvendige rammeverk for å avhjelpe de ønskede forbedringer.

7.3.5.1.1 Rammen for Software Acquisition Capability Maturity Model

Denne modellen: "Software Acquisition Capability Maturity Model"[SA-CMM] er utviklet ved Software Engineering Institute ved Carnegie Mellon Universitetet.

Bakgrunnen for at de laget en slik modell var at de tidligere har laget en tilsvarende modell for utvikling av software, "Capability Maturity Modell for Software"^[SW-CMM]. Antagelsen var at denne måten å tenke på var direkte anvendbar når de skulle lage en modenhetsmodell for anskaffelse. Ideen er da at SW-CMM beskriver utviklerens rolle mens SA-CMM beskriver kjøpers rolle i software anskaffelsesprosessen.

SA-CMM begynner anskaffelsen med prosessen å definere brukerkrav. Anskaffelsesorganisasjonen kan starte opp noen av aktivitetene før et prosjekt etableres. SA-CMM inkluderer noen aktiviteter forut for kontrakt, slik som det å lage forespørsel og etablering av krav til dokumentasjon. I

^[SA-CMM] Software Acquisition Capability Maturity Model (SA-CMM) Version 1.01, SEI-96-TR-020 ESC-TR-96-020, desember 1996

^[SW-CMM] Capability Maturity Model SM for Software, Version 1.1, SEI-93-TR-024 ESC-TR-93-177, Februar 1993

systemdefinisjonsfasen er de to modellene parallelle. SA-CMM avsluttes når kontrakten for leveranse av software produkter og eventuelt tilhørende tjenester er avsluttet.

Selv om SA-CMM logisk er i samsvar med SW-CMM, er de to modellene ikke noe speilbilde av hverandre. Systemutvikling og systemanskaffelse er to forskjellige prosesser med forskjellige mål og forskjellige aktiviteter. Derfor er de sentrale elementene i SA-CMM forskjellige fra SW-CMM.

Denne forskjellen betyr ikke at de to modellene ikke er kompatible. Modellene har den samme arkitektur, design og format og vurderingsmetoder. De to modellene kan brukes parallelt i et prosjekt hvis det anses nødvendig. De to modellene med fokus på de samme krav, med samme fremdrift og med den samme målsetting – høy kvalitet, kan bruket sammen gi en synergieffekt.

SA-CMM identifiserer nøkkelprosesser for fire av de fem nivåene av modenhet. Disse nøkkelprosessene oppgir mål som må nås for å tilfredsstille det angitte nivå av modenhet. På denne måten er dette en modningsprosess hvor fremgang måles i steg eller trinn. De beskrevne nivå angir en kjøreplan for å oppnå et høyere nivå på modenhetsskalaen.

I CMM er det slik at en del av målene og aktivitetene i en nøkkelprosess er tilfredsstilt eller utført på et lavere nivå. Men en nøkkelprosess er slik at den kan ikke tilfredsstilles uten at *alle* målsettinger er oppnådd. Et modenhetsnivå er slik definert at graden er nådd når man behersker alle nøkkelprosesser.

Når man når opp til et høyere modenhetsnivå krever modellen at alle mål fra de lavere nivå vedlikeholdes. På denne måten er trinnene på stigen komplementære.

SA-CMM er laget for å være generisk nok til bruk i enhver offentlig eller industribedrift, uavhengig av størrelse eller type software som skal anskaffes. Ved bruk av modellen i en gitt organisasjon kan tilpasning være nødvendig og på samme måte kan det være nødvendig med skreddersøm for en gitt anskaffelse.

SA-CMM kan anvendes på anskaffelse av alle typer software (embedded og stand-alone) samt der COTS skal anskaffes. I slike tilfeller kan det være nødvendig med skreddersøm.

Det ligger ingen begrensninger i anvendelsen av SA-CMM – den er derfor ment å være egnet bruk gjennom softwarens hele livsløp. I livsløpet til et system kan det være nødvendig med mange anskaffelser, det kan være i utviklingsfasen eller i driftsfasen. Hvis det er nødvendig med oppgradering eller systemtilpasninger i driftsfasen, kan driftsorganisasjonen ta på seg rollen til anskaffelsesorganisasjonen. Slik er modellen ment å være anvendbar også i denne fasen.

Et prosjekt må ha definerte basislinjer for software ytelseskrav, for kostnad og for tidsplan til anskaffelsesprosjektet.

I SA-CMM er det forstått at prosjektledere historisk sett ikke har hatt den nødvendige handlefrihet for å kunne styre etter disse tre parametre og dermed oppnå et så optimalt resultat som mulig. Derfor har modellen inkludert muligheten for at prosjektledelsen kan justere en av basislinjene for å bedre kunne styre og gjøre trade-offs. Dette er gjort for å øke sjansen for en vellykket anskaffelse.

7.3.5.1.2 Arkitekturen til SA-CMM

SA-CMM definerer fem modenhetsnivå. Hvert modenhetsnivå (unntatt nivå 1) er angitt med visse kapabiliteter og er beskrevet med noen sentrale prosessaktiviteter. Disse sentrale prosessaktivitetene er angitt med mål og har noen felles egenskaper. Disse felles egenskapen kan sees på som attributter som

angir hvorvidt implementering og institusjonalisering av de sentrale prosessene er effektiv, om dette vil kunne gjentas og bli en varig prosess.

De fem felles egenskaper gjennomgås her:

Løfte om å gjennomføre²⁷. Et løfte om å yte beskriver de handlinger organisasjonen må gjennomføre for å etablere prosessen og forsikre seg om at den kan videreføres. Løfte om å yte vil typisk si å etablere en policy i organisasjonen og fremskaffe en sponsor i ledelsen.

Evne til å gjennomføre²⁸. Evne til gjennomføre beskriver de elementene som må være på plass på forhånd i prosjektet eller organisasjonen for å innføre software anskaffelsesprosessen på en forsvarlig måte. Evne til å gjennomføre vil typisk si at man har oversikt over ressurser, etablert en organisasjonsstruktur og at det er etablert opplæring.

Aktivitet gjennomført²⁹. Aktivitet gjennomført beskriver de roller og prosedyrer som trengs for å implementere en sentral prosessaktivitet. Aktivitet gjennomført vil typisk si etablerte planer og prosedyrer, utført arbeide, monitorere dette og ta nødvendige beslutninger i ledelsen.

Målinger og analyser³⁰. Målinger og analyser beskriver behovet for å måle prosessen og analysere disse måleresultatene. Målinger og analyse vil typisk si at man for eksempel ser på målinger for å vurdere status og effektivitet av de aktiviteter man gjennomfører.

Verifisere anvendelse³¹. Verifisere anvendelse beskriver handlinger for å forsikre seg om at handlingene gjennomføres i tråd med beskrevet prosess. Verifisere anvendelse vil typisk si at man gjennomfører reviews i ledelsen.

7.3.5.1.3 Modenhetsnivå i SA-CMM

Anskaffelsesorganisasjonen må øke involvering, lederskap og disiplin for å kunne øke til et høyere nivå av modenhet. I fortsettelsen her beskrives de fem modenhetsnivå i SA-CMM, og de viktigste prosessforbedringer på hvert nivå blir nevnt.

- **Initielt** – Softwareanskaffelsesprosessen har karakter av å være ad-hoc, og til tider svært kaotisk. Svært få prosesser er definert og suksess avhenger helt og holdent av individuelle ferdigheter og bestrebelser. For at en organisasjon skal bli mer modent enn dette initielle nivået, må det innføres grunnleggende kontrollrutiner for å fremme selvdisciplin.
- **Repeterbart** – Grunnleggende software anskaffelsesprosjektrutiner er etablert for å planlegge alle deler av anskaffelsen, håndtere software krav, følge prosjektteamet og leverandørenes ytelse, håndtere prosjektets kostnads- og tidsmilepæler, evaluere produkter og tjenester samt suksessfull overføring til driftsorganisasjonen. Prosjektteamet reagerer stort sett på hendelser fra omgivelsene når de oppstår. Den nødvendige disiplin er til stede for å gjenta tidligere suksesser i prosjekter i tilsvarende anvendelsesområder. For at en organisasjon skal modnes ut over dette nivået av selvdisciplin, må den ta i bruk veldefinerte

²⁷ Commitment to perform

²⁸ Ability to perform

²⁹ Activities performed

³⁰ Measurement and analysis

³¹ Verifying implementation

prosesser som fundament for forbedring.

- Definert- Anskaffelsesorganisasjonens software anskaffelsesprosess er dokumentert og standardisert. Alle prosjekter bruker en godkjent, tilpasset versjon av organisasjonens standard software anskaffelsesprosess for å fremskaffe sine softwareprodukter og tjenester. Prosjekt og merkantil ledelse er proaktiv, og forsøker å forutse og håndtere hendelser fra omgivelsene før de skjer. Risikohåndtering er en del av alle deler i prosjektet og organisasjon gjennomfører opplæring på alt personell som er involvert i anskaffelsen. For at en organisasjon skal modnes ut over dette nivået hvor prosessene er definert, må den la beslutninger om prosessen og produktene hvile på kvantitativt grunnlag slik at objektivitet kan brukes og rasjonelle beslutninger taes.
- Kvantitativt – Detaljerte malinger på software anskaffelsesprosessen, dens produkter og tjenester samles inn. Softwareprosessen, produktene, og de samlede tjenester er kvantitativt forstått og kontrollert. For at en organisasjon skal modnes ut over dette nivå, må det være etablert en kontinuerlig forbedringsprosess
- Optimaliserende- Fortløpende prosessforbedring muliggjøres av kvantitativ feedback fra prosessen og ved å prøve ut nye ideer og teknologi. På høyeste nivå vil en organisasjon erkjenne at kontinuerlig forbedring (og endring) er nødvendig for å overleve.

7.3.5.1.4 Grunnleggende prinsipper for hvordan SA-CMM skal forstås.

SA-CMM er en generisk modell laget for bruk i mange organisasjoner. I et gitt tilfelle vil implementering og forbedring måtte variere. SA-CMM er en modell for å forbedre organisasjonen. SA-CMM fokuserer på å bygge prosessmodenhet inn i en organisasjon.

Aktiviteten i SA-CMM beskriver typiske trekk man normalt vil finne i en software anskaffelsesprosess på hvert modenhetsnivå. SA-CMM foreskriver ikke hvordan en organisasjon skal forbedre sine prosesser, men den beskriver normative praksis for organisasjoner på fem nivå av modenhet. Modenhetsnivåene kan sees på som et system for å prioritere sine prosessforbedringsaktiviteter.

SA-CMM beskriver hovedprosessområder og aktiviteter, men er ikke utfyllende, og andre prosessområder og aktiviteter kan tenkes. SA-CMM tar ikke for seg alle de faktorer som får følger for et software anskaffelsesprosjekt. Eksempler på deler som ikke er tatt med er systems engineering, mennesker i prosessen og teknologi.

Kvaliteten til et softwareprodukt eller tjeneste er slik det sees under SA-CMM stort sett gitt av den softwareprosess som er anvendt for å utvikle, anskaffe eller vedlikeholde den. En hver prosess, hevdes det i CMM, kan forbedres; kontinuerlig prosessforbedring er en nødvendighet for å forbedre effektivitet og vedlikeholde konkurransefortrinn i skiftende omgivelser.

Det er ikke bare én “rett vei” eller bare én ”golden path” for å implementere en software anskaffelsesprosess. SA-CMM er ikke hugget i stein. Derfor – med noen få omhyggelig utvalgte unntak, pålegger SA-CMM ikke hvordan en software anskaffelsesprosess skal implementeres eller hvem som skal gjøre en gitt handling. Det som beskrives er de karakteristikker som software anskaffelsesprosessen skal ha.

SA-CMM er teknologiuavhengig. Det er ikke foreskrevet noen spesielle verktøy, metoder eller teknologi. Det som anses nødvendig av verktøy, metoder og teknologi bør gjøres tilgjengelig for å understøtte prosessen.

Profesjonell avdømming må brukes når man skal tolke aktiviteter etter SA-CMM for en spesiell organisasjon. SA-CMM skal tilpasses organisasjonen; det er ikke organisasjonen som skal restruktureres for å avspeile SA-CMM.

Kompetansen til de personene som gjør arbeidet er en avgjørende faktor for prosjektgjennomføring og suksessfulle organisasjoner. (Kompetanse forstås her som kunnskap om anvendelsesområdet, software anskaffelsesmetoder, kvantitative metoder og det å inneha interpersonlige evner samt evne til å løse problemer

7.3.5.2 Presentasjon av modellen

Nivå	Fokus	Nøkkelprosesser
5 Optimaliserende	Kontinuerlig prosess forbedring	Styrt utvikling av anskaffelser Kontinuerlig prosessforbedring
4 Kvantitativt	Kvantitativ ledelse	Kvantitativ ledelse av anskaffelsen Kvantitativ prosessledelse
3 Definert	Prosess standardisering	– Opplæringsprogram Risikostyrt anskaffelse Håndtering av kontraktsytelse Kontroll med prosjektytelse Prosessdefinerings og vedlikehold
2 Repeterbart	Grunnleggende prosjektledelse	Overføring til drift Evaluerings Kontraktsadministrasjon og oversikt Prosjektledelse Kravutvikling og kravhåndtering Behandling av forespørsler Planlegging av softwareanskaffelse
1 Initielt	Kompetente enkeltindivider og helter	

Tabell 15 Softwareanskaffelse - Dugelighets- og modenhetmodell

7.3.5.3 Anvendelse av en modenhetsmodell

Hvorfor skal vi ha enda en modell? Etter å ha arbeidet med prosjekt K2IS Blokk 1 samtidig som jeg har skrevet denne oppgaven har jeg gjort mine observasjoner. Når jeg nå skal skrive om anskaffelsesprosess vil jeg derfor naturlig nok relatere dette til de erfaringer jeg har gjort meg selv.

Dette kapitlet var ment som en enkel men helhetlig presentasjon av en anskaffelsesprosess slik jeg ser den bør gjennomføres. Dette har hengt over meg som en mare i svært lang tid, uten at det har vært mulig å spore særlig fremgang. Hvorfor har dette vært så vanskelig? En anskaffelsesprosess i en organisasjon som den jeg jobber i til daglig er en svært sammensatt prosess. Jeg har jobbet der en god stund nå, og har fått et godt innblikk i hvordan prosessen gjennomføres. Men jeg har ikke noe helhetlig bilde. Til det er antallet aktører og parallelle prosesser i dag for mange. Så hva er alternativet til å presentere en "helhetlig prosess". Jeg mener at basert på ideene i modenhetsmodellen SA-CMM bør vi definere hvor vi er, og hvor vi vil. Vi bør beskrive de nøkkelprosesser vi i dag har, sammen med de mest sentrale aktører og interessenter. Deretter bør vi skissere mål for hvor vi vil ha forbedringer og deretter gjennomføre disse.

7.3.5.3.1 Forslag til målbilde

Målbilde settes til nivå 3. Dette er det nivået hvor prosessen har elementer i seg til å bli noe mer enn enda en form for dokumentasjon av de tingene alle vet hva de burde ha gjort.

Slik jeg ser det er dette et langsiktig mål. For å komme i gang trengs det noen grunnleggende elementer. Disse er listet i all sin korthet her:

- Overordnet modell
- Identifisering av Nøkkelprosesser
- Modellering av nøkkelprosesser i dag med fokus på forbedringer
- Opplæring i nøkkelprosesser
- Bruk av nøkkelprosesser

Ved å ta tak i disse i en plan for forbedring, vil det være mulig å oppnå læring i organisasjonen.

På noe lengre sikt vil det være mulig med noen utvidelser. Disse må vurderes mot kostnader og nytteverdi før de innføres i stor skala. To mulig utvidelser skisseres her:

- Identifisering av nøkkeltall – forberedelse for målbare prosesser
- Kontraktsevaluering – studier og sammenligning

Disse to utvidelsene vil kunne bringe organisasjonen videre til neste nivå hvis det er ønskelig.

7.3.5.3.2 Anbefaling

Jeg har i dette kapitlet gjennomgått SA-CMM. Slik jeg ser det er dette en nyttig modell for å beskrive modenhetsnivå i en organisasjon med hensyn til software anskaffelser.

Jeg har i presentasjonen av modellen vært tro mot ordet software, og gjentatt dette – kanskje til det kjedsommelige. Slik jeg ser det er dette ikke avgrenset til software anskaffelser – det bør og kan helt sikkert utvides til å dekke også andre typer anskaffelser.

Jeg anbefaler at vi i vår organisasjon tar i bruk SA-CMM som et kvalitetsforbedringsverktøy.

Jeg anbefaler at vi setter oss årvisse mål for hvor langt vi ønsker å komme, basert på en visjon om å komme på nivå 3 i løpet av tre til fem år. Jeg legger inn en viss usikkerhet, fordi jeg i dag ikke kjenner alle de elementene som bestemmer modenhetsstatus på organisasjonen. Normalt sies det å ta minst to til tre år for å gå fra et nivå til neste. Jeg vil ikke være bastant på hvorvidt vi i dag er på nivå én eller to – bare konstatere at det er svært stor forskjell mellom det vi gjør best og det vi gjør mindre bra.

7.4 OPPSUMMERING

Her oppsummer jeg de forslag jeg har kommet med og de største utfordringene dette vil medføre for organisasjonen, med fokus på gevinstrealisering.

7.4.1 ERFARINGER ETTER FASE 1 I PROSJEKTET

Gjennom deltakelse i prosjekt K2IS Blokk 1 har jeg gjort meg enkelte erfaringer.

Det var mange ubesvarte spørsmål etter fase 1 i K2IS Blokk 1. Disse spørsmålene skulle blitt brukt til å forbedre prosess, og lette videreføringen. En pause oppsto mellom fase 1 og fase 2. Det var hele tiden planen at man skulle kjøre videre så raskt som mulig. Forskjellige forhold gjorde dette umulig. Med en klar formening om hvordan prosessen mellom kunde og leverandør skulle være ville oppstarten godt og vel et år senere gått mye mer smertefritt. Det ble formulert en del spørsmål knyttet til nettopp dette, og jeg reflekterer her litt over disse. Det vil være riktig å vurdere om min besvarelse i noen grad tar tak i disse problemstillinger.

Det ble gjennomført en samtale mot slutten av K2IS Blokk 1, fase 1 basert på et dokument skrevet av Ove Scheel for CCIS House. Dette er noen av de hovedpunkter jeg satt igjen med etter denne samtalen.

7.4.1.1 Modell for utvikling og samhandling

Var alle parter inneforstått med hva det innebærer å kjøre prosjekter etter en evolusjonær modell?

En evolusjonær utviklingsprosess for K2IS' innebærer til dels radikalt nytt i forhold til konvensjonelle prosjektmodeller. Ingen av partene hadde særlig erfaring med dette. Selv om noen av partene har kjørt inkrementelle prosjekter med delleveranser tidligere, så er dette noe annet. Uttalelser som: "Dette er jo ikke noe nytt, slik har jo vi kjørt prosjekter i alle år" er farlige fordi de avspeiler en holding der man ikke har åpenhet til at det kanskje er noe nytt – til dels uprøvd i miljøet – som faktisk må læres og forstås.

Skal det være noen vits i å kjøre etter en ny prosess så må vi gjøre det fordi vi er overbevist om at dette gir oss de gevinstene som loves.

Hovedforskjeller til konvensjonelle prosesser:

- kortere og mindre inkremerter
- flere Leveranser
- krav endres og justeres underveis
- hyppig tilbakemelding fra virkelige brukere
- aktiv deltagelse fra brukermiljøet
- detaljplaner utvikles kun for de neste nærmeste Iterasjonene
- planer justeres kontinuerlig underveis – typisk etter hver Iterasjon
- tidsestimater justeres etter hvert Iterasjon
- kostnadsestimater justeres etter hvert Iterasjon
- alle planlagte Iterasjoner behøver ikke gjennomføres
- kontinuerlig måling og prosessforbedring

Disse teknisk-administrative elementene nevnt her medfører at kvalitetssikring må utføres på annen måte enn for konvensjonelle prosjekter. Dette vil bety merarbeid for partenes kvalitetssikringsavdelinger og kanskje også utløse motstand i disse miljøene.

Dette vil også medføre at kontrakter og avtaler må utformes på annen måte enn for konvensjonelle prosjekter. Stikkord her er: betalinger, garantier, formelle milepæler, overtagelsestidspunkter, lisenser, eierskap, endringsordrer blant annet.

Den organisasjonen som ble prøvd ut i K2IS hadde mange aktører. Dersom ikke interessekonflikter mellom partene og byråkrati i forbindelse med kommunikasjon mellom partene håndteres på en skikkelig måte, så vil dette være en alvorlig trussel mot gjennomføring av et slikt prosjekt i praksis. Momenter her er alt fra tilgang til adekvate sluttbrukere til integrasjon av Leveranser som består av moduler fra forskjellige underleverandører

7.4.1.2 Kompetanse hos aktørene

Kompetanse og kunnskap hos nøkkelpersonell om en ny prosess er avgjørende for lyktes. Dersom kompetansen ikke er til stede, så må denne fremskaffes. For det meste av nøkkelpersonellet betyr dette ulike former for opplæring av det personell som tildeles disse rollene. For prosjektleder(e) kan dette like gjerne løses gjennom ansettelse eller innleie av personer med dokumentert kompetanse, og også helst erfaring med evolusjonære prosesser. Dette vil i tillegg kunne ha den positive tilleggseffekt at det kommer inn nøytrale personer.

7.4.1.3 Prosess- forbedring og oppfølging

Ansvaret for å sørge for at prosessen virkelig følges kan kanskje delegeres, men i og med at kunden har den antatt største gevinsten av at den følges, må kunden også ivareta sine interesser ved å følge opp at det virkelig skjer. Det bør vurderes å etablere en eller annen form for kontrollinstans en styringsgruppe e.l. som jevnlig gransker prosjektet og kontrollerer at prinsippene etterleves. Dette gjøres gjennom granskinger og inspeksjoner av det som produseres i prosjektene.

7.4.2 RESULTATER BENYTTET VIDERE I K2IS BLOKK 1 FASE 2

Fra K2IS Blokk 1, fase 1 ble det utført mye godt arbeid. Ikke alle resultatene er blitt videreført til selve utviklingsprosjektet, kalt K2IS Blokk 1, fase 2. En av de ting som ikke er videreført direkte er resultatene fra Wbs 1.2, beskrevet her i forrige kapittel. Arbeidet her pekte direkte på hvilke – til dels dramatiske – endringer som måtte gjennomføres både hos leverandør og kunde for at det man kalte en evolusjonær prosess skulle kunne gjennomføres. En evolusjonær prosess som beskrevet i kapittel 6.2 krever visse endringer på begge sider i forhold til dagens praksis. Dagens prosess er i stor grad en fossefallsorientert prosess, med islett av elementer fra en mer inkrementell prosess, dog i en svært forenklet form. Dette er ikke ment som noen entydig kritikk av valget om ikke å innføre en evolusjonær prosess. Den endring som evolusjonær prosess må medføre, er det ikke gitt at man ville kunne gjennomført, all den tid dette ikke ble etablert før prosjektet ble startet opp med en ny fase. Å innføre store endringer i prosessen samtidig som prosjektet kjører, viste seg å være svært lite fruktbart i fase 1. De elementene som det er verdt å trekke frem viser dog at diskusjonen om evolusjonær utvikling har båret noen frukter.

7.4.2.1 Milepælsdrevet

Prosjektet er i delt inn med tre hovedmilepæler. Hovedmilepæl 1 etter 12 mnd, Hovedmilepæl 2 etter 18 måneder og Hovedmilepæl 3 etter 26 måneder. Fremdriften i prosjektet måles månedlig med statusmøter mellom kunde og leverandør. Dette månedlige fokus har nå etter at hovedmilepæl 1 er passert blitt ytterligere forsterket, Det vil nå bli gjennomført månedlige integrasjonstester, sammenstillinger og leveranser.

Hovedmilepæl 1 har hatt et fokus på infrastruktur. Dette kan ligne mye på en arkitekturleveranse i UP. Dette er et sterk risikoreduserende tiltak, og derfor svært positivt i forhold til hva som kunne vært alternativet hadde man valgt en mer tradisjonell prosess. Siden leveransen også er en fysisk leveranse til kundens testmiljø, der også enkelte sluttbrukertester skal gjennomføres må dette kunne anses som en første leveranse. Milepæl 2 skal være en leveranse med utstrakt funksjonalitet til sluttbrukermiljøet. Intensjonen er da at tilbakemeldinger fra sluttbrukermiljøet skal bidra til å gjøre endelig leveranse så nær forventningene som mulig. I den grad det er mulig vil nye oppdateringer tilflytte sluttbrukermiljøet i perioden mellom milepæl 2 og 3. Det er dog en utstrakt skepsis til å gjøre for store endringer i denne fasen da dette kan komplisere opplæringen.

7.4.2.2 Brukerdrevet

Brukergrupper er etablert og skal være en sterkt integrert del av utviklingen. Systemet tre hoveddeler har hver sin brukergruppe, med deltakere fra sluttbrukeren av systemet. Disse vil nå spille en sentral rolle i utformingen av brukergrensesnitt og forhåpentligvis også i prioriteringen av funksjonalitet.

7.4.2.3 Etablering av helhetlige suksessfaktorer

For å fremme et helhetlig resultat ble det forsøkt etablert en ansporingsmekanisme. Denne var ment å fungere slik at oppfylles et sett av gjennomgående egenskaper på totalsystemet, ville leverandøren få en ekstra belønning. I mangel på spesifikke krav til de enkelte delene og en ufullstendig systemarkitektur som viser hvordan delementene inngår i en større helhet har man etablert et sett med scenarier. Disse scenariene tar for seg de mest kritiske prosessene som systemet skal understøtte. Ansporingen skulle påvirke leverandøren til fokusere som et totalsystem og sikre at systemet som helhet opptrer tilstrekkelig godt. Slik jeg oppfatter det ble disse ansporingsmekanismene ikke en del av endelig kontrakt. Det kan likevel bære aktuelt å vurdere ved en senere anledning.

8 EVALUERING

Kapittelet er det siste i denne hovedoppgaven som forsøker å si noe til oppgavens formulering, hvordan store organisasjoner kan anskaffe systemer som virker sammen over tid. Kapitlet er todelt, der del én ser på hvordan oppgaven svarer på kravene stilt i kapittel 2-krav, del 2 vurderer hvordan min egen organisasjon har tatt dette i bruk og jeg omtaler dette som egen organisasjons modenhet.

8.1 EVALUERING AV OPPGAVENS BESVARELSE AV KRAVENE

8.1.1 INNLEDNING

Hensikten med dette kapitlet er å se på relasjonen mellom det som er stilt som krav i kapittel 3 - og det som er foreslått og beskrevet som løsninger i denne oppgaven.

Dette kapitlet angir om oppgaven svarer på kravene som tidligere er gitt.

Dette kapitlet tar utgangspunkt i kapittel 3 og gjennomgår alle senere kapitler med kapittel 3 som et bakteppe. Kapittel 3 tok for seg kravene til systemutvikling og interoperabilitet. Oppgavens tittel er "Hvordan kan store organisasjoner anskaffe informasjonssystemer som virker sammen over tid?" Store organisasjoner har altså en oppgave med å anskaffe informasjonssystemer som skal virke sammen, og dette skal de gjøre også etter hvert som tiden går.

Inndelingen blir da som i kapittel 3 – systemutvikling og interoperabilitet. Systemutvikling sees i sammenheng med store organisasjoner, der begrepet anskaffelsesmetode er vel så dekkende.

8.2 KRAVET TIL SYSTEMUTVIKLING

I denne oppgaven er systemutvikling brukt som begrep i litt forskjellige sammenhenger. Termen system er vist å være alt fra mindre tekniske enheter, via større tekniske system til store tekniske system i samvirke med brukende organisasjon.

Her vil systemutvikling være alle de aktiviteter som gjennomføres for å endre virkeligheten og skape noe nytt. Systemutvikling er en samhandling mellom utviklende organisasjon (leverandør) mottakende organisasjon (kunden) og systemeier (forvaltning)

Kravet til systemutvikling, slik det er gitt i denne oppgaven er beskrevet i kapittel 3.

I denne oppgaven er systemutvikling definert som en aktivitet under systemmodell og anskaffelsesmodell. Kravet til systemutvikling under systemmodell er håndtert under kapitelet 3.4.2. Systemutvikling i anskaffelsesmodellen er et resultat av definerte mål i anskaffelsesprosessen.

I denne oppgaven er det vist hvordan kundens forventninger kan innfries gjennom

- Involvering av kunden gjennom prioritering av funksjonelle krav i anskaffelsesfasen
- Involvering av kunden gjennom raske leveranser i anskaffelsesfasen, et bidrag til å redusere opplevd usikkerhet

8.3 KRAVET TIL INTEROPERABILITET

Interoperabilitet oppleves i dag som et mål i seg selv. Vi er i dag til forskjell fra noen år tilbake omgitt av artikler og sitater hvor dette begrepet brukes flittig.

Jeg har i denne oppgaven valgt å bygge på etablerte oppfatninger, basert på den forståelse som har vært kommunisert i NATO gjennom de siste tiår. Interoperabilitet er altså ikke noe nytt som er oppdaget gjennom innføring av begrepet Nettverksbasert Forsvar – NBF. NBF er mer en konsekvens av det arbeidet som har vært langt ned nasjonalt og i alliansen i over et tiår. Interoperabilitet er nå definert som et kombinert krav til prosedyrer – les menneskelig samhandling og krav til tekniske systemer og standardisering av grensesnitt.

Interoperabilitet en del av de krav som er stilt i beskrivelsen av virksomhetens krav til det tekniske system.

Videre er det tekniske system bygget opp av enheter med definerte grensesnitt.

I denne oppgaven er det vist hvordan krav til interoperabilitet kan oppnåes gjennom

- Definerte krav til samvirke fra beskrivelse av virksomheten

Denne beskrivelsen utvikles kontinuerlig og er basis for utvikling av egen organisasjon, så vel som basis for å beskrive krav til de underliggende tekniske system.

- Definerte tekniske systemenheter med definerte grensesnitt

Det tekniske system er beskrevet som en portefølje av enheter. Hver enhet er beskrevet med funksjonelle egenskaper så vel som gjennom definert grensesnitt. Enhetene settes sammen til komplette tekniske system i en konfigurasjon for å oppnå en bestemt hensikt.

8.4 KRAVET TIL SYSTEMUTVIKLING OVER TID

I denne oppgaven er systemutvikling definert som en aktivitet under systemmodell og anskaffelsesmodell. Kravet til systemutvikling under anskaffelsesmodell er håndtert i kapittel 3.4.2.

Dette betyr at vi har en kontinuerlig utvikling av systemet (arven) som et resultat av behov for endringer i organisasjonen og systemforvaltning av det tekniske system.

I denne oppgaven er det vist hvordan kundens forventninger kan innfries gjennom

- Involvering av kunden gjennom kontinuerlig definering av kontekstmodell – K2S

I denne oppgaven har jeg vist hvordan endringer i utviklingsorganisasjon skal håndteres gjennom:

- Etablering og vedlikehold av en modenhetsmodell for organisasjonen

8.5 HVORDAN KAN STORE ORGANISASJONER ANSKAFFE INFORMASJONSSYSTEMER SOM VIRKER SAMMEN OVER TID?

Denne hovedoppgaven har hatt som målsetting å besvare hvordan store organisasjoner kan anskaffe informasjonssystemer som virker sammen over tid.

8.5.1 EN SAMMENSTILLING AV BESVARELSEN I OPPGAVEN

I denne oppgaven har jeg drøftet systemutvikling, konfigurasjonsstyring, interoperabilitet og arkitektur.

Dette er elementer i en kontinuerlig endringsprosess.

Jeg har i denne oppgaven vist hvordan store organisasjoner kan anskaffe informasjonssystemer som virker sammen over tid gjennom

- En definert systemforvaltningsmodell
- En definert anskaffelsesmodell
- En definert beslutningsmodell
- En definert forbedringsmodell

8.5.2 BESVARELSEN SETT OPP MOT EGNE KRAV

Kapittel 3 ble oppsummert med en tabell som viste de elementer som en helhetlig systemanskaffelsesprosess må håndtere. Her anviser jeg hvor de enkelte elementene er håndtert i mitt modellapparat.

	Håndteres	Hvordan håndtere endring:	Beskrevet i:
Lover og regler	Ja	N/A	Beslutningsmodell
Struktur	På en åpen måte	Synliggjøre konsekvenser	Systemmodell/ Anskaffelsesmodell
Økonomi	Alternative løsninger	Synliggjøre konsekvenser	Anskaffelsesmodell
Organisasjon	På en åpen måte	Holde à jour (Enkelt å oppdatere)	Systemmodell
Arbeidsprosesser	På en åpen måte	Holde à jour (Enkelt å oppdatere)	Systemmodell

Brukerkrav	På en åpen måte	Synliggjøre konsekvenser	Anskaffelsesmodell/ Systemmodell
Risiko	På en åpen måte	Synliggjøre konsekvenser og tiltak	Anskaffelsesmodell/ Beslutningsmodell
Systemer	På en åpen måte	Holde à jour (Enkelt å oppdatere)	Systemmodell
Interoperabilitet	På en åpen måte	Holde à jour (Enkelt å oppdatere)	Systemmodell
Teknologi	På en åpen måte	Synliggjøre konsekvenser	Systemmodell

Tabell 16 Krav besvart i oppgaven

Håndtering av endringer over tid ligger primært i forbedringsmodellen, og en endring medfører at de andre modellene endres/oppdateres og at nytt opplegg for opplæring etableres i forbedringsmodellen.

9 KONKLUSJON - VIDERE ARBEID

I dette avsluttende kapittelet forsøker jeg å konkludere. Deretter kommer jeg med et forslag til hvordan jeg ville gått videre, med det jeg anser for å være det neste skritt hvis mine resultater skulle anvendes i egen organisasjon.

9.1 KONKLUSJON

Jeg har i denne oppgaven tatt for meg problemstillingen; hvordan kan store organisasjoner utvikle systemer som virker sammen over tid?

Jeg har valgt et område som jeg kjenner relativt godt til, som del av mitt arbeid i Forsvarets Logistikkorganisasjon. Området jeg benytter som anvendelsesområde er operative IKT systemer i Forsvaret. Med min bakgrunn har det primært vært Hæren som er min referansebakgrunn.

Jeg har hatt en serie med prosjekter som startet opp i 1999, utvikling av K2IS til Hæren som case for denne oppgaven. Disse prosjektene er nå i hovedsak avsluttet, eller i ferd med å avsluttes. Nye prosjekter som vil videreutvikle K2IS i Forsvaret er under etablering i FLO.

Hvis jeg nå ser tilbake på det vi har forsøkt å gjøre siden starten i 1999, og vi skulle gjøre dette på nytt – hva ville jeg anbefale?

9.1.1.1 Vår ambisjon

For lesere som ikke kjenner min rolle, kan det være på sin plass å presisere denne. Jeg har i tilknytning til disse prosjektene hatt en jobb som fagansvarlig. Dette vil si at jeg er tiltenkt en rolle knyttet til forvaltning og drift av et endelig system. I gjennomføringen av prosjektet har jeg hatt oppgaver knyttet til metode for systemutvikling og arkitektur. Jeg har bare i liten grad deltatt i daglig oppfølging av prosjektet, hva angår utvikling, test eller innføring og opplæring av brukere.

Det vi forsøkte å gjøre var å fremskaffe et system. Det vi hadde var en omfattende kommunikasjonsinfrastruktur som skulle brukes av det nye systemet.

Det ble samtidig etablert en strategisk samarbeidsavtale med CCIS House, en sammenstilling av blant annet de bedrifter som så langt hadde laget kommunikasjonsinfrastrukturen. CCIS House hadde gjennom nesten to år studert fenomenet K2IS til Hæren, og ble i 1999 tildelt en kontrakt om å levere et system.

Samtidig ble det innført arkitektur som metode for å styre den videre utviklingen. Jeg tror det er flere grunner til at dette ble gjort. Ledelsen vår i FLO, representert ved Erik Hammer var en grunn. Han var overbevist om at dette var riktig. Noe av grunnen til dette var nok at FFI hadde gjennomført en grundig studie av KKI-HÆR. Der ble arkitektur som metode fremhevet som et av flere virkemidler for at Forsvaret skulle lykkes med en slik omfattende anskaffelse. Videre hadde bedriftene i CCIS House gjennom sitt studiearbeid fremskaffet kunnskap om det som var gjort i USA, se omtale av C4ISR-AF. Sist men ikke minst var det mange enkeltmennesker som gjennom sin kompetanse og erfaring var kommet frem til at arkitektur hadde mange lovende egenskaper. Jeg var da en av disse.

9.1.1.2 Resultat

Hæren har i dag et operativt K2IS, kalt NORTaC-C2IS. Dette er levert av de prosjektene som her er omtalt. Systemet fungerer, og er i operativ bruk. Jeg kan ikke i denne oppgaven gå i detalj i å beskrive styrker og svakheter til systemet. Dette er som man kanskje forstår ikke offentlig informasjon. Det jeg kan snakke om er hvordan arkitektur som metode har fungert i forbindelse med dette arbeidet. Dette vil bli en subjektiv vurdering, som jeg ikke forventer at alle andre involverte skal være enig i.

Når det gjelder arkitektur som metode vil si at vi ikke har oppnådd særlig gode resultater. Jeg ser dette da i lys av at vi har brukt relativt mye ressurser til å få det til. Slik jeg ser det har vi i dag ikke oppdaterte arkitekturbeskrivelser av virksomheten og systemet NORTaC-C2IS.

Modellene som ble etablert tidlig i prosjektet, er ikke holdt ved like, og oppdaterte modeller har derfor ikke vært en del av leveransen fra Leverandøren. Vi hadde en klar oppfatning av hva vi kunne oppnå i form av kontroll hvis vi hadde anvendt arkitektur som metode. Den valgte modellen for gjennomføring ga oss ikke oppdaterte virksomhets- og systembeskrivelser.

9.1.1.3 Hva burde vært oppnådd?

Det burde vært etablert gode gjennomgående rutiner for kjøring av K2IS prosjekter, i tillegg til et funksjonelt system.

I 1999 var det en klar forståelse for at de første prosjektene var de første av en serie prosjekter. Det var klart for alle parter at dette var noe man kom til å gjøre mange ganger. Det var hos alle aktører en relativt god forståelse for forvaltning av materiell, slik at konfigurasjonsstyring, materiellkontroll etc. var innarbeidet som begrep.

Etablering av gode gjennomgående forvaltningsrutiner så som konfigurasjonsstyring og endringshåndtering for det softwaretunge K2IS burde derfor blitt etablert. I dette innbefattes i lys av denne oppgaven; virksomhetsmodeller og systemmodeller herunder grensesnittbeskrivelser. I dette legger jeg også gjennomgående samhandlingsrutiner mellom sluttbruker, driftspersonell, forvaltning og leverandør for å underbygge kontinuerlig drift som basis for videreutvikling. I tillegg burde man gjennom en slik serie med prosjekter opparbeidet en viss kompetanse i kostnadsanalyse, for eksempel basert på USE-case. Sist men ikke minst burde man få et system i operativ drift, som i hovedsak dekket de mest sentrale funksjonelle behov til sluttbrukeren.

9.1.1.4 Hva ble oppnådd?

Det ble fremskaffet et godt teknisk system, men svært lite ble gjort for å etablere gode gjennomgående rutiner for kjøring av K2IS prosjekter.

Det er fremskaffet et system i operativ drift NORTaC-C2IS. Dette dekker de mest sentrale funksjonelle behov for sluttbrukeren.

Det ble videre gjort et godt arbeid i å beskrive samhandlingsrutiner mellom sluttbruker, driftspersonell og delvis forvaltning. Dette arbeidet kalt "Managementprosessen", basert på ITIL³² er dessverre bare

³² The Information Technology Infrastructure Library (ITIL®) er et rammeverk basert på "best practice" som har til hensikt å fremme leveranser av IT-tjenester av høy kvalitet. ITIL består av et omfattende sett av ledelsesprosesser laget for å støtte virksomheter i å oppnå både høy økonomisk gevinst samt nytte av sine IT tjenester. Prosessene er leverandøruavhengige og har blitt utviklet for å kunne være en veiledning på tvers av infrastruktur, utvikling og operativ bruk av IT., <http://en.wikipedia.org/wiki/ITIL>, link benyttet 2007-04-29

delvis satt ut i praksis. Dette skyldes omorganiseringen i Forsvaret, der Hærens etablerte praksis er satt på vent i påvente av FK-KKIS skal etablere én modell i Forsvaret.

Det er også etablert en omfattende personlig erfaring hos aktørene i forvaltning og industri. Dette er i liten grad forsøkt dokumentert, slik at det kan danne grunnlag for læring i organisasjonen.

9.1.1.5 Hva ble ikke oppnådd?

Grunnlag for gjennomføring av ytterligere komplekse K2IS-prosjekter er ikke på plass.

Det kan ikke sies at vi i dag har oppdaterte virksomhetsbeskrivelser, der det fremgår hvilke elementer som er understøttet av dagens K2IS. Det kan heller ikke sies at vi har oppdaterte systemmodeller, der det tydelig fremgår konfigurasjonselementer og konfigurasjonsstyrte grensesnitt. Det er ikke etablert gjennomgående forvaltningsrutiner for K2IS. Med dette mener jeg for eksempel et gjennomgående konsept for feilrapportering, og brukerinvolvering i konfigurasjonsstyring av K2IS.

Det vil si at den basis som er etablert, bare til en viss grad er egnet som basis for videreutvikling.

9.1.1.6 Hvordan kunne dette vært unngått?

Primært burde forvaltningsaspekter ved fremskaffelse av K2IS vært høyere prioritert. Med utgangspunkt i min besvarelse burde prosjektstyring i større grad hatt en iterativ og inkrementell tilnærming. Det kunne vært gjort ved etablering av gode tilbakekoblingssløyfer tidlig i prosjektet.

Videre burde de virksomhetsbeskrivelser som var utgangspunkt for prosjektet i mye større grad vært vedlikeholdt, ved at dette ble ansett som viktig – men ikke minst at de menneskene som hadde utført jobben opprinnelig ble dratt med videre i prosessen.

Videre burde systemmodeller med grensesnittsbeskrivelse vært tydelig beskrevet som del av leveransene. Dette burde vært en del av den konfigurasjonsstyring som forvaltningen tidlig skulle ha fokus på å få på plass.

9.1.1.7 Hva bør vi gjøre annerledes neste gang?

Et K2IS er et komplekst system – dette krever også noe kompleks ledelse og styring.

Kompleksiteten i det K2IS som vi forsøker å fremskaffe burde tilsi en god forståelse for at dette er komplekst. Arkitektur som metode er en metode for håndtere kompleksitet. Jeg mener fortsatt at arkitektur som metode er riktig for å oppnå vår ambisjon om å utvikle systemer som virker sammen over tid. Jeg mener også at arkitektur som metode vil gjøre vår jobb i FLO enklere. Hva bør vi så gjøre? Mitt svar er kanskje litt pretensiøst på egne vegne, men jeg mener svaret er å ta den medisin som her er foreskrevet i denne oppgaven, altså helhetlig systemanskaffelse.

En balansert tankegang med det modellapparatet som her er beskrevet, vil slik jeg ser det gi de ønskede resultater. En slik tilnærming starter på toppen i måten vi etablerer prosjektet. Helhetlig styring vil kunne gjennomføres med innføring av min beslutningsmodell. Dette er en bevisstgjøring om de beslutninger som tas knyttet til prosjektet. En naturlig følge av dette er å organisere seg i henhold til denne modellen, slik at aktørene får makt til å gjennomføre det modellen foreskriver.

En systemmodell, er en bevisstgjøring om det som skal anskaffes i prosjektet.

En anskaffelsesmodell, er en bevisstgjøring av den måten vi organiserer gjennomføring av prosjekter, her kanskje med et særlig fokus på arkitektur som metode.

Sist men ikke minst er en forbedringsmodell en bevisstgjøring av at vi må og skal evaluere oss selv kontinuerlig.

9.1.1.7.1 Beslutningsmodell

En beslutningsmodell sier noe hvilke beslutninger som må taes og av hvem.

Måten dette bør gjennomføres på er i seg selv kjernen i løsningen. Etableringen av en modell for samhandling må begynne i det små med få aktører. De involverte aktører må identifiseres, motiveres og læres opp i god samhandling. Den største utfordringen for oss i FLO har vært, og er at ikke alle egne aktører forstår kompleksiteten i det som forsøkes oppnådd. Når ting går galt har det vist seg at det veldig lett å skyldte på "den andre parten". FLO må som forvaltning ha klare linjer knyttet til iverksetting, kontroll, oppfølging og avslutning av prosjekter. En enkel modell kan være fristende fra et kontrollperspektiv, men sikrer ikke at kompleksiteten håndteres. Jeg foreskriver en noe mer kompleks modell, som også involverer vår kunde i Forsvaret – de operative brukerne.

9.1.1.7.2 Systemmodell

En systemmodell sier noe om hva som skal anskaffes.

De grunnleggende prinsippene bør etableres først og stabiliseres. I de tidlige fasene er i tillegg til funksjonelle analyser og demonstratorer også arkitekturmodeller helt grunnleggende. Det var i gjennomføringen av prosjektene til Hærens K2IS en stor grad av fomling og usikkerhet knyttet til hva som burde utarbeides av arkitekturmodeller, hvordan disse skulle utarbeides, hva de skulle brukes til osv. I den samme periode valgte FLO å fokusere på videreutvikling av MACCIS, med påfølgende opplæring. Dette ble gjort uten at de grunnleggende prinsippene på tilsvarende måte var implementert som del av samhandlingen med leverandør. Det jeg mener å si var at det bare i liten grad var etablert klare krav til leveranser av arkitekturmodeller fra leverandøren. Når vi da et stykke ut i byggeprosessen anmodet om å få dette på plass, ble dette naturlig nok ansett som støy og ekstra arbeid i et prosjekt som allerede var godt i gang med å lage et system.

9.1.1.7.3 Anskaffelsesmodell

En anskaffelsesmodell sier noe om hvordan samhandling skal skje i gjennomføring av prosjektet.

Det hadde vært mulig å etablere flere sykluser i løpet av en prosjektperiode. Hver syklus kunne hatt et gjennomløp som endte i en ny leveranse. Dette kom etter hvert delvis på plass, blant annet på grunn av press fra kundesiden som ønsket årlige innføringsløp av nye leveranser. Bevisstgjøringen av sammenheng mellom leveranseprosjekt som leveranser, og den samtidige etablering av gode rutiner for forvaltning burde i mye større grad vært fokuser tidlig i prosjektet. Konfigurasjonsstyring av det som var levert, mulighet for å endre fokus underveis ville i større grad vært mulig med en mer tydelig iterativ og inkrementell tilnærming i prosjektgjennomføringen.

9.1.1.7.4 Forbedringsmodell

En forbedringsmodell sier noe om hvordan læring skal gjennomføres i organisasjon.

Samtidig som de tre modellene her kalt beslutningsmodell, systemmodell og anskaffelsesmodell var under etablering, burde det vært etablert et regime for kontinuerlig forbedring. Vi fikk tidlig en leveranse fra SINTEF kalt Experience Well (erfaringsbrønn). Bare navnet i seg selv klinger bra. Her var det flere konkrete tiltak til hvordan dette kunne stables på bena. Rapporten har et særlig fokus på å etablere, bruke, erfare og gjenbruke arkitekturmodeller. Prinsippet om erfaringslæring kan også anvendes på alle de andre elementer av helhetlig systemanskaffelse.

9.1.1.8 Oppsummering

Jeg har her forsøkt å oppsummere de erfaringer jeg har gjort meg med helhetlig systemanskaffelse knyttet til de konkrete prosjektene i Hærens K2IS. Vår ambisjon var omfattende og i etterpåklokskapens klare lys kanskje ikke helt realistiske. Vi ønsket både å fremkaffe et nytt system som ikke var anskaffet før. Vi ønsket å etablere nye rutiner for forvaltning som vi ikke hadde fått til før. Vi ønsket å gjøre dette med en utvalgt leverandør, hvor forvaltningens samhandling ikke nødvendigvis var basert på åpen konkurranse. Dette ønsket vi mens å gjøre samtidig som vi innførte arkitekturmodellering som teknikk og metode.

Det som kanskje særlig ble frustrerende var at vi ikke klarte å etablere et godt regime hvor arkitekturmodellering inngår som et grunnleggende prinsipp.

For egen regning vil jeg si at den læring som enkeltindivider fikk i denne prosessen er brakt med videre. I etableringen av nye prosjekter har noen prinsipper om helhetlig styring blitt lagt til grunn. Arkitektur som støtte i beslutningsprosessene er under etablering i FLO/IKT som har innført et arkitekturråd som rådgivende organ for beslutninger. Dette kan bety at vi er ferd med å ta noe av den medisin som her er foreskrevet.

9.2 VIDERE ARBEIDE – INNFORING I EGEN ORGANISASJON

Denne oppgaven har basert på behovet for å anskaffe systemer som virker sammen over tid, foreslått en modellbasert, arkitekturdrevet og evolusjonær systemanskaffelsesprosess. Dette ble i kapittel 7 satt inn i en ramme der jeg fra et overordnet nivå predikerer en formell beslutningsmodell samtidig som jeg angir at egen organisasjons modenhet bør monitoreres kontinuerlig.

Slik jeg ser situasjonen nå vil det neste skrittet, hvis jeg skulle fortsatt denne oppgaven være å nettopp vurdere modenheten i egen organisasjon.

Jeg anbefaler derfor at det gjennomføres en evaluering av egen organisasjons status for å avdekke mangler sett i lys av denne oppgaven. Når dette er gjennomført vil det være mulig å komme med forslag til tiltak for forbedringer.

9.2.1 EGEN ORGANISASJONS MODENHET

Min egen organisasjon har vært i omstilling gjennom flere år. Forsvarets Logistikkorganisasjon ble vedtatt opprettet av gjennom St prop 55 (99/00)^[55-99/00].

I følge den modellen som ble skissert her er organisasjonen nå i fase 3, og i ferd med å etablere en mer prosessbasert internstruktur for FLO. Dette oppfattes mest som et forsøk på å få fjernet den gamle struktur og kultur som var etablert i de grenvise forsyningskommandoer. Prosessen som ble satt i gang har så langt endt i en ny organisasjon som ble etablert 1. september 2005. Dette etter at denne datoen var forskjøvet to ganger.

En ny organisasjon under etablering må nødvendigvis få en viss tid på å sette seg i stand til å utføre sine oppgaver. Det er derfor vanskelig å snakke om modenhet i denne organisasjonen. Jeg velger likevel å ta opp modenhet i organisasjonen til vurdering. Dette fordi jeg mener at etableringen av en ny organisasjon kunne hatt en modenhetstilnærming både til omstilling, etablering og ikke minst fremtidig styring av ny organisasjon. Med dette mener jeg at ved å tenke modenhet, vil man bevisst måtte tenke status i dag, bevisst måtte sette seg mål for hvor man ønsker at man vil komme, og ikke minst bevisst legge en strategi eller plan for å aktivt forbedre seg.

9.2.1.1 Avgrensning

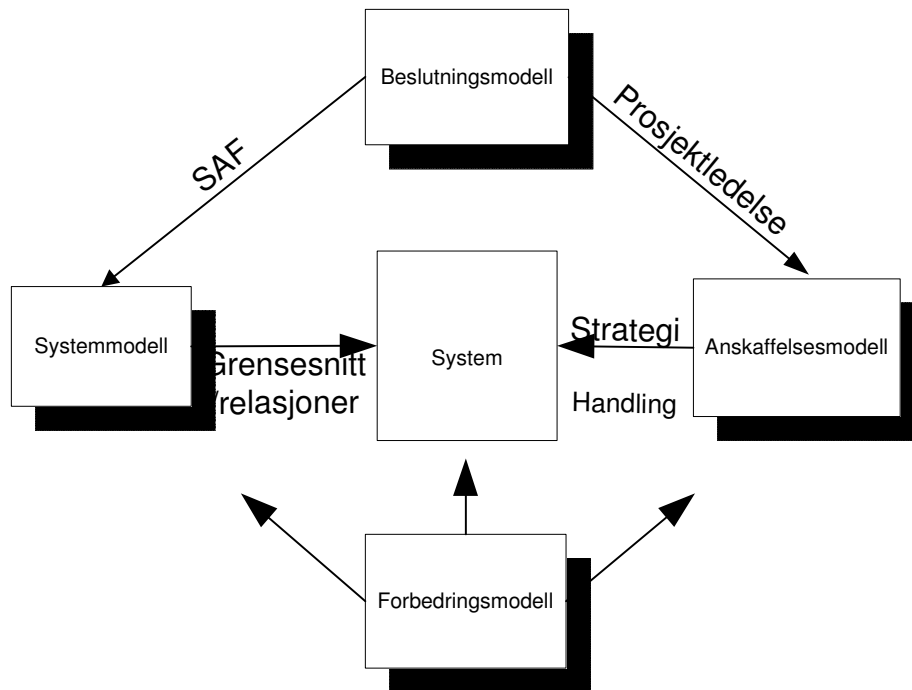
Jeg velger å fokusere på enkelte utvalgte områder som avgrenser seg til anskaffelse av enkelte typer materiell. Spesifikt er dette anskaffelse av informasjonssystemer. Dette er videre avgrenset til informasjonssystemer som skal brukes til å understøtte operativ virksomhet – her benevnt K2IS. Dette innebærer at beslutningsstøttesystemer som i nye FLO/IKT omfatter både administrative og operative informasjonssystemer bare delvis er dekket. Denne avgrensningen skyldes primært at jeg ikke kjenner til organisasjon eller organisasjonskultur og kompetansemiljø som i dag forvalter de administrative støttesystemer. Det er dermed ikke sagt at min analyse kan være interessant også for disse miljøene, men dette er ikke tilsiktet.

9.2.1.2 Utkast til En Undersøkelse

Jeg foreslår å gjennomføre en undersøkelse av modenheten i forhold til mine 3 foreslåtte modeller. Dette kan danne grunnlag for et videre arbeid knyttet til hver av de tre modellene, men også til forståelse av læring i organisasjonen.

9.2.1.2.1 Modellapparat

Det foreslåtte modellapparat er ment å dekke alle relevante aspekter av hvordan en organisasjon som FLO kan anskaffe interoperable K2IS som virker sammen over tid.



9.2.1.2.2 Forståelse og bruk av Systemmodell

- Dagens praksis

Systemmodellen i Forsvaret i dag knyttet til K2IS er under etablering. To forhold er grunnleggende for forståelsen av dagens situasjon. Det ene forholdet er arven, det vil si de eksisterende systemer, deres tilknyttede brukerorganisasjoner og ikke minst den arv som er knyttet til de tidligere grenvise forvaltningene av systemer.

Det andre forholdet er knyttet til modellering og arkitektur. Begge disse forhold bør avdekkes og analyseres.

- Bruk av modell

Fokus kan i første omgang rettes til den del av systemporteføljen som er ansett å være relevant for videreføring.

- Forslag til tiltak

Tiltakene utformes i samarbeide med systemansvarlig forvaltning innenfor K2IS, som i dag er FLO/IKT. Jeg vil anbefale å vektlegge de positive observasjoner knyttet til dagens bruk av modellering.

9.2.1.2.3 Forståelse og bruk av Anskaffelsesmodell

- Dagens praksis

Forsvaret har i dag en vel etablert anskaffelsesmodell. Denne modellen vurderes mot det som her er anbefalt som en modellbasert, arkitekturdrevet og evolusjonær systemanskaffelsesprosess

- Bruk av modellen

Det bør undersøkes om det er spesielle problemer knyttet til modeller som underlag for kontrakter.

Bruk av flere leverandører tidlig i systemanskaffelsesprosessen håndteres spesielt.

Forhold som hindrer bruk av systemintegrator for å håndtere deler eller hele systemintegrasjonsrollen til hovedaktør vurderes spesielt.

- Forslag til tiltak

Tiltakene utformes i samarbeid med FLO/Investeringsavdelingen som er ansvarlig for anskaffelser i Forsvaret.

9.2.1.2.4 Forståelse og bruk av Beslutningsmodell

- Dagens praksis

Beslutninger følger i vår organisasjon flere formater. Det kan vurderes om Balansert Målstyring, en metode som er innført skal vurderes spesielt.

Dagens beslutningsmodell knyttet til system- og anskaffelsesprosessen er i seg selv omfattende med svært mange aktører. Det bør her vurderes å sette ned en gruppe som får et mandat til å utrede dette separat.

Temaet kan oppfattes som følsomt, ikke minst fordi beslutninger berører samhandling med eksterne aktører som FK KKIS, FST og FD.

Tidlig involvering med disse aktørene er derfor avgjørende for undersøkelsens godhet.

- Bruk av modellen

Her bør bruk av formelle metoder og verktøy vektlegges. Arkitektur som basis for beslutninger bør vektlegges da dette er den best ansette basis for å håndtere komplekse problemstillinger.

- Forslag til tiltak

Tiltakene utformes i tett samarbeid med den eksterne aktører som har deltatt i studien.

9.2.1.2.5 Forståelse og bruk av Forbedringsmodell

- Dagens praksis

Gjennom denne undersøkelsen vil det være mulig å etablere et grunnlagt for fremtidig forbedring. Undersøkelsen kan få svært forskjellige resultater avhengig av hvor man starter. Jeg vil anbefale å ta for meg en mindre del av organisasjonen. Undersøkelsen bør gjøres sammen med Human Resource personell, som formelt har ansvaret for organisasjonsstruktur og kompetanse. Det vil også være på sin plass å invitere inn representanter fra tillitsmannsapparatet til å delta i dette arbeidet. Det bør vektlegges at personellet er den viktigste ressurs for en forvaltningsorganisasjon som FLO.

- Bruk av modellen

Organisasjonens evne til kontinuerlig forbedring bør undersøkes i detalj.

- Forslag til tiltak

Tiltakene bør utarbeides med fokus på organisasjonens evne til kontinuerlig forbedring.

9.2.2 OPPGAVENS RELEVANS I FORHOLD TIL EGEN ORGANISASJON

Hvis jeg selv skulle vurdere relevansen av denne oppgaven for min egen organisasjon, FLO, ville jeg sett på om det var mulig å gjennomføre en slik undersøkelse.

Hvis det er gjennomførbart, og det med en viss grad av sikkerhet kunne anslås at elementer av det som foreslått kan forbedre organisasjonen, vil jeg si at oppgaven er relevant.

Jeg mener at jeg i denne oppgaven fremmer flere forslag til hvordan vår egen organisasjon kan forbedre oss og bli mer profesjonell på flere områder. De angitte områdene knyttet til beslutninger, systemforvaltning, anskaffelser og evnen til kontinuerlig forbedringer er slik jeg ser det grunnlag for at Forsvaret som en stor organisasjon kan anskaffe systemer som virker sammen over tid.

10 REFERANSER

[FLO] Flo, Forsvarets logistikkorganisasjon, ble opprettet for å samle all logistikkvirksomhet for Forsvaret i én organisasjon, <http://www.mil.no/flo/start/om/>, link benyttet 2007-04-29

[ST-prp42] St.prp. nr. 42 (2003-2004) Den videre moderniseringen av Forsvaret i perioden 2005-2008, Tilråding fra Forsvarsdepartementet av 12. mars 2004, godkjent i statsråd samme dag. (Regjeringen Bondevik II), <http://odin.dep.no/repub/03-04/stprp/42>, link benyttet 2007-04-29

[FK-KKIS] Forsvarets kompetansesenter for kommando og kontroll informasjonssystemer, <http://www.mil.no/fkkis/>, link benyttet 2005-04-25

[PROG97] Direktiv for programområdet K2IS i Hæren, Utgitt 1997, Generalinspektøren for Hæren

[Systbesk] TADKOM Teknisk systembeskrivelse, brosjyre utgitt av Hærens Forsyningskommando

[TADKOM] TADKOM Taktisk områdesamband for Felthæren, brosjyre utgitt av Hærens Forsyningskommando

[MRR] Deltamobile MRR, produktark utgitt av NFT Ericsson Communications ANS

[IIS-bok] Understanding Information Infrastructure, Ole Hanseth and Eric Monteiro, Manuscript 27. Aug., 1998, kapittel 3 Defining information infrastructures, <http://heim.ifi.uio.no/~oleha/Publications/bok.3.html#pgfId=913144>, link benyttet 2005-05-01

[St-mld22] St.meld. nr. 22, (1997-98), Hovedretningslinjer for Forsvarets virksomhet og utvikling i tiden 1999-2002, Tilråding fra Forsvarsdepartementet av 26. februar 1998, godkjent i statsråd samme dag, <http://www.regjeringen.no/nb/dep/fd/dok/regpubl/stmeld/19971998/Stmeld-nr-22-1997-98-/6.html?id=191482>, link benyttet 2007-04-29

[LANDOPS] Forsvarets doktrine for landoperasjoner, Forsvarsstaben, 2004, http://www.mil.no/multimedia/archive/00054/Forsvarets_doktrine_54469a.pdf, link benyttet 2007-04-29

[MIP] Multilateral Interoperability Programme, <http://www.mip-site.org>, link benyttet 2007-04-29

[Sommer97] Sommerville, Ian & Sawyer, Pete, Book, Requirements Engineering, A good Practice Guide, ISBN 0 471 97444 7

[SikkLov] Lov om forebyggende sikkerhetstjeneste (sikkerhetsloven). Ikrafttredelse: 2001-07-01, (Erstattet da Datasikkerhetsdirektivet, Forsvarets Overkommando/Sikkerhetsstaben 1998,) [html](#), link benyttet 2005-05-01

[EvalPR99] Bjørn W. Bjanger, Tor Neple, Jan Øyvind Agedal, Arne-Jørgen Berre, Evaluering av programmet K2IS i Hæren, ISBN 82-14-01220-1, utgitt 1999-04-19

[GJFPL99] Johnsen, Roger, Gjennomføringsplan for konseptfasen, prosjekt FP 5910 Feltfunksjonsapplikasjoner, datert 1999-09-07.

[PTG,98] Future NATO Interoperability model, Cand Scient hovedoppgave, Institutt for Informatikk, Universitetet i Oslo, May 1998.

[Praha02] Prague Summit Declaration, Heads of State and Government participating in the meeting of the North Atlantic Council, Prague on 21 November 2002, <http://www.nato.int/docu/pr/2002/p02-127e.htm>, link benyttet 2005-05-01

[St.pr-42] St.prp. nr. 42 (2003-2004), Den videre moderniseringen av Forsvaret i perioden 2005-2008, 12. mars 2004, <http://odin.dep.no/fd/norsk/dok/regpubl/stprp/010001-030030/dok-bn.html>, benyttet 2006-01-07.

[NNEC] NATO C3 Technical Architecture (NC3TA), Appendix D. NATO Network Enabled Capability (NNEC) Feasibility Study Strategic Recommendations, <http://194.7.80.153/website/book.asp?menuid=15&vs=0&page=volume5%2Fapd.html>, link benyttet 2007-04-29

-
- [CCRP-99] Command and Control Research Symposium, June 1999, Naval War College, Newport, Rhode Island, CD-ROM, Track 5: Interoperability Coalition and Doctrine, Architecture: The Road to Interoperability, Dr. Raymond J. Curts, CDR, USN (Ret.), Strategic Consulting Inc., Dr. Douglas E. Campbell, LCDR, USNR-R (Ret.), Syneca Research Group, Inc.
- [DCI-99] High level steering group (HLSG), Defense Capability Initiative, Annex1 to HLSG-N(99)6, Rev 2, 18 Oct 99
- [FFI-3429] Teknologi, Forsvar og forsvarstrukturer, FFI rapport - 2000/03429, Solstrand, Ragnvald H, september 2000.
- [ARF] Regelverk for anskaffelser til Forsvaret (ARF), Det kongelige forsvarsdepartement, 01 09 2004, <http://odin.dep.no/fd/norsk/dok/regelverk/010051-990014/dok-bn.html>, referanse benyttet 2005-01-08
- [Boehm] A Spiral Model of Software Development and Enhancement, Barry W. Boehm, Computer, May 1988 (Vol. 21, No. 5) pp. 61-72, <http://doi.ieeecomputersociety.org/10.1109/2.59>, link benyttet 2007-04-29
- [IEEE1220] IEEE 1220, Trial-Use Standard for Application and Management of the Systems Engineering Process, februar 1995, <http://www.software.org/quagmire/descriptions/ieee1220.asp>
- [UP] Unified Process, From Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Unified_Process
- Søk KvalMe Google søk, Utført 2006-08-20, Søkord: kvalitative metoder dataverktøy, Treff 593, URL: <http://www.google.no/search?hl=no&q=kvalitative+metoder+dataverkt%C3%B8y&btnG=S%C3%B8k&meta=>
- [IN-MMO] IN-MMO: Modelling med objekter, <http://www.ifi.uio.no/~mmo/>
- [Catalysis] Objects, Components, and Frameworks with UML: The Catalysis(SM) Approach (The Addison-Wesley Object Technology Series), <http://www.catalysis.org/books/ocf/index.htm>
- [RM-ODP] Reference Model - Open Distributed Processing, ISO 10476 og ITU-T X.900
- [NC3TA] NATO C3 TECHNICAL ARCHITECTURE, Version 2.0 [Dec 15, 2000] - ISSC NATO Open Systems Working Group, <http://194.7.79.15/>
- [DSTC] DSTC RM-ODP Information Service, http://archive.dstc.edu.au/AU/research_news/odp/ref_model/
- [C4ISR-AF] Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework Version 2.0, 1997-12-18, URL: <http://www.fas.org/irp/program/core/fw.pdf#search=C4ISR-AF>, link benyttet 2007-04-29
- [C4ISR-FR] C4ISR Architecture Working Group Final Report, 14 April 1998, http://www.c3i.osd.mil/org/cio/i3/AWG_Digital_Library/pdffdocs/fnlrprt.pdf
- [AAP-6] Allied Administrative Publication - 6 "NATO Glossary Terms and Definitions
- [PTG,98] Future NATO Interoperability model , Cand Scient hovedoppgave, Institutt for Informatikk, Universitetet i Oslo, May 1998.
- [NIPD] APP - 10: NATO Interoperability Planning Document (Hvilken datering gjelder på den kopien jeg har fått at PTG?)
- [OPER] Operative informasjonssystemer. Kartlegging og samordningsbehov. WINJUM Eli, HEDENSTAD Ole-Erik, SLETTEN Geir, FFI- 2000.
- [LISI] Levels of Information Systems Interoperability (LISI), C4ISR Architecture Working Group, 30 mars 1998, <http://www.sei.cmu.edu/isis/guide/introduction/lisi.htm>, link benyttet 2007-04-29.
- [MIP] Multilateral Interoperability Programme, <http://www.mip-site.org>, link benyttet 2007-04-29.
- [O Ha 1] Ole Hanseth, Eric Monteiro and Morten Hatling. (1998)., Developing information infrastructure: The tension between standardization and flexibility, Science, Technology and Human Values. Vol. 21 No. 4, Fall 1996, 407-426, <http://www.ifi.uio.no/~oleha/Publications/sthv.html>, link benyttet 2007-04-30
-

-
- [PIFO-96] Felles prosess- og informasjonsmodell for Forsvaret, PIFO, sluttrapport, Forsvarets Overkommando, 1996.
- [IS-direk] FORSVARSJEFENS DIREKTIV FOR INFORMASJONSSYSTEMER (FSJ IS-DIREKTIV), oktober 1997
- [FD-retn] OVERORDNEDE RETNINGSLINJER FOR INFORMASJONSSYSTEMER I FORSVARET, Forsvarsdepartementet, mai 1997
- [ARGUS] Kopi av forsiden på Forsvarets IntraWEB for prosjekt ARGUS, med kopier av WWW-sidene for IS/IT.
- [IKT-POL] Forsvarsdepartementet, Policy for militær tilpasning og anvendelse av informasjons- og kommunikasjonsteknologi Forsvaret, 2005, http://www.regjeringen.no/upload/kilde/fd/bro/2006/0020/ddd/pdfv/277246-ikt_policy.pdf, link benyttet 2007-04-30
- ARF Anskaffelsesregelverk for Forsvaret, <http://odin.dep.no/fd/norsk/dok/regelverk/010051-990014/dok-bn.html>, link benyttet 2007-04-29
- KONSIF Nytt konsept for investeringsstyring i Forsvaret, 09.09.04, <http://odin.dep.no/fd/norsk/dok/regelverk/010101-990015/dok-bn.html>
- [LOA] FAD, Lov om offentlige anskaffelser, 2001-07-01, <http://www.lovdata.no/all/nl-19990716-069.html>, link benyttet 2007-04-29
- [ST-PRP1] St.prp. nr. 1 (2005-2006), <http://odin.dep.no/fd/norsk/dok/regpubl/stprp/010001-030039/dok-bn.html>
- [INVSTY] Nytt konsept for investeringsstyring i Forsvaret, Forsvarsdepartementet, 09.09.04, <http://www.regjeringen.no/upload/kilde/fd/reg/2004/0039/ddd/word/220003-investeringskonsept.doc>, link benyttet 2007-04-29
- [PRINSIX] Forsvarets fellesprogram i prosjektledelse med sertifisering og studiepoeng, <http://www.prinsix.no/prinsix>, link benyttet 2007-04-29
- [MACCIS] MACCIS – Minimal Architecture for CCIS in the Norwegian Army versjon 2, Report Part A and B, datert 2001-12-15, SINTEF Report no A: STF40F01067 B: STF40F01068
- [OBJ-AN93] Matiassen, Munk-Madsen, Nielsen, Stage – Objektorientert Analyse, 1993, ISBN 87-7751-065-8
- [UML] Universal Modeling Language, UML Consortium (1997a) Object Management Group. UML Consortium (1997b) Rational Software Corporation., UML Consortium (1997c) Rational Software Corporation
- [APP6A] APP6-A Army Publication no 6 A, Military symbols for land based systems
- [AschGyld] Aschehoug og Gyldendals STORE NORSKE LEKSIKON, 1983 4.opplag.
- [LEFFING] . D. Leffingwell, D. Widrig, and E. Yourdon, Managing Software Requirements: A Unified Approach: Addison Wesley, 1999, http://books.google.com/books?vid=ISBN032112247X&id=h4pPpXp-xrEC&pg=PA376&lpg=PA376&ots=g1PpQ_t96x&dq=Requirements+management+can+be+defined+as+a+systematic+approach+to+eliciting&sig=Ak2MpVYZH0zbpEBVW1i0sedD8b4, link benyttet 2007-04-30
- [Unif Pr] I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process: Addison-Wesley, 1999
- [IFPUG] The International Function Point Users Group” (IFPUG) – <http://www.ifpug.org/> link benyttet 2007-04-30
- [St-prp-1] St.prp. nr. 1, (2005–2006) FOR BUDSJETTÅRET 2006, Utgiftskapitler: 1700–1795, Inntektskapitler: 4700–4799, Tilråding fra Forsvarsdepartementet av 23. september 2005, godkjent i statsråd samme dag. (Regjeringen Bondevik II), <http://www.statsbudsjett.dep.no/2006/dokumenter/html/fagdep/fd/kap04.htm>, link benyttet 2007-04-30
- [55-99/00] St.prp. nr. 55 (1999-2000) Forsvarets logistikkfunksjoner, Framtidig virksomhet, styring og organisering. Tilråding fra Forsvarsdepartementet av 5. mai 2000, godkjent i statsråd samme dag.